DOCUMENT RESUME

ED 260 702 IR 011 779

TITLE	Proceedings of the Annual Conference on "The Role of the Computer in Education" (5th, Arlington Heights, Illinois, February 20-22, 1985).
INSTITUTION	Micro-Ideas, Glenview, IL.
PUB DATE	Feb 85
NOTE	252p.
PUB TYPE	Collected Works - Conference Proceedings (021) -- Viewpoints (120) -- Reports - Descriptive (141)
EDRS PRICE	MF01/PC11 Plus Postage.
DESCRIPTORS	Computer Assisted Instruction; *Computer Literacy; Computer Software; *Curriculum Development; Elementary Secondary Education; Higher Education; *Instructional Innovation; *Microcomputers; Programing; Skill Development
IDENTIFIERS	*Computer Uses in Education

ABSTRACT

The 46 papers in this proceedings summarize the work of academic and private groups which seek to provide a means of integrating the utilization of the computer into an established curriculum; descriptions of sample courses are included. The contents include: (1) Four Precollege Computer Curricula: A Symposium; (2) Data Processing Management Association; (3) Computer Science for Secondary Schools: Course Content; (4) The Report of the ICTM SC3; (5) Assessing Library Microcomputer Needs; (6) Word Processors: "Ghost Writer"; (7) Telecommunications for Educators--An Introduction; (8) Structured BASIC on the Macintosh; (9) Issues and Trends; (10) Computers in Biological Education; (11) BASIC for Students with a Limited Mathematics Background; (12) Word Processing: Introduction in the Computer Lab; (13) Microworlds and Macroworlds; (14) Professional Training in CBE: The View of One University; (15) Kinder Keys: A Keyboarding Program; (16) Lasagna Code--An Alternative to Spaghetti; (17) Software Evaluation Is a Personal Affair; (18) Computer Expedition 1984: What to Do When the Novelty Is Gone; (19) Effects of Microcomputer Technology on Young Handicapped Children; (20) The Role of the Microcomputer at Illinois State University; (21) Polishing the Apple in Vocational Education; (22) Keyboarding in the K-12 Curriculum; (23) The Microcomputer Lab of the Future; (24) Taking the Next Step--Slowly; (25) CAI--Is That All There Is?; (26) Outlining a Practical First Computer Course for Special Educators; (27) College Chemistry Graduates Have Poor Communication Skills; (28) Computer Power: A Problem Solving Course for All Levels; (29) Pascal as a Programming Language for Beginners; (30) Word Processing Is Good for Elementary Teachers and Students; (31) Keyboarding; (32) Teachers Talk--The Failure of Computers in Education?; (33) Three Anchors and Computer-Assisted Learning; (34) Increasing Female Involvement in the Use of Computers; (35) Using New Technologies in Education: A Redefinition; (36) Child Tracking System; (37) Computer Literacy: Still an Issue or Should We Change Its Name?; (38) Improving Instruction through Appropriate Software for Special Ed; (39) Taking the Byte out of Counseling; (40) Sweetening Your Apple; (41) The Computer as a Tool in the Junior High Classroom; (42) Implementation of a K-8 Computer Curriculum; (43) CIC--Computers in the Curriculum; (44) Excellence in Administration: Planning and Presenting; (45) Advanced Techniques of Logo Programming; and (46) Athletic Administration: Sport Statistics. (JB)

PROCEEDINGS

ED260702

U.S. DEPARTMENT OF EDUCATION
NATIONAL INSTITUTE OF EDUCATION
EDUCATIONAL RESOURCES INFORMATION CENTER (ERIC)

☒ This document has been reproduced as received from the person or organization originating it.

☐ Minor changes have been made to improve reproduction quality.

• Points of view or opinions stated in this document do not necessarily represent official NIE position or policy.

MICROCOMPUTERS
MICROCOMPUTERS

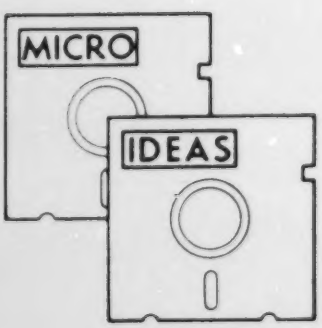
MICRO IDEAS

THE ROLE OF THE COMPUTER IN EDUCATION 5

MICRO IDEAS

MICROCOMPUTERS
MICROCOMPUTERS

IR011779



"PERMISSION TO REPRODUCE THIS MATERIAL HAS BEEN GRANTED BY

Richard F. Nelson

TO THE EDUCATIONAL RESOURCES INFORMATION CENTER (ERIC)."

2701 CENTRAL ROAD GLENVIEW, IL 60025

PROCEEDINGS

OF

THE FIFTH ANNUAL

THE ROLE OF THE COMPUTER IN EDUCATION

SPONSORED BY

MICRO-IDEAS

2701 CENTRAL ROAD

GLENVIEW, ILLINOIS 60025

312-998-5065

HELD AT

THE ARLINGTON PARK HILTON

ARLINGTON HEIGHTS, IL

FEBRUARY 20, 21, & 22ND

1985

MICRO-IDEAS is a microcomputer consortium located in Glenview, Illinois. Originally a Title IV C project, MICRO-IDEAS is now supported by membership fees.

RICHARD F. NELSON, DIRECTOR
BRUCE GJERTSEN
KAY HICKS
JEAN PERCY
CECIL PRETTY
JUDY SANDER
PENNY SHATTUCK

MICRO-IDEAS ADMINISTRATIVE DISTRICT

Glenview Community Consolidated School District 34
DR. WILLIAM J. ATTEA, SUPERINTENDENT

ADVISORY COMMITTEE MEMBERS

MR. DONALD ROGAN, DISTRICT 203, CHAIRMAN
MR. DONALD ADCOCK, DISTRICT 41
DR. ROBERT BORTNICK, DISTRICT 59
MRS. PAULA FILLIMAN, DISTRICT 34
MR. BRUCE JAMES, DISTRICT 203
MR. BERNARD J. KELLY, DISTRICT 214
MRS. MARY ELLYN MADDEN, DISTRICT 181

ALTERNATE MEMBERS

MRS. ANN DANA, DISTRICT 181
MR. LEO HUNDT, DISTRICT 214
MR. THOMAS POWERS, DISTRICT 34
MR. JOHN TOFANO, DISTRICT 59

PUBLISHED APRIL, 1985 BY MICRO-IDEAS, 2701 CENTRAL ROAD, GLENVIEW, IL 312/998-5065

FOREWORD

The Role of the Computer in Education 5

MICRO-IDEAS proudly presents the proceedings for the 1985 conference. We hope that this years conference provided some immediate relevance for those attending. This might have been some new software to use, or a different approach to take to an old problem. We hope that there will be some long term relevance as well. This might take the form of goal setting and coordinating the use of the computers within your building or district. If either happened to you, then the conference was a success.

Another goal for the conference was to impress upon all the rapidly changing nature of the technology. While planning is imperative for the success of the computer in the school, flexibility may be equally prized. The rise of the technology occurred because of the freedom and power to the individual. Educators should foster the creativity and individual growth that the technology can provide. The students need to be guided in their exploration.

This conference was the result of great effort by the staff of **MICRO-IDEAS** and tremendous cooperation and support from all attendees and vendors. We look forward to arranging for the Role of the Computer in Education VI on February 12, 13, 14, 1986. Make your plans now!

CURRICULUM DEVELOPMENT

Planning and organizing for effective, thorough education is the goal for most educators. The influx of computers into the curriculum has caught many unawares. Some thought them a fad, others felt that the new technology would not affect them. The plans presented here represent data processing, engineering and mathematics. These groups have perhaps been the most amenable to adapting to the use of computers. It is hard to think of some of their activities (e.g. banking, space exploration, etc.) not being done via computer.

The following documents represent the attempt to bring orderly involvement of the new technology. Each of these groups seeks to provide a means to integrate the computer into the curriculum from their unique viewpoint. Only when each subject discipline has come to grips with the topic for itself will there be a possibility for the true synthesis and integration of the computer as a tool of inquiry, rather than as an object of instruction.

TABLE OF CONTENTS

1. FOUR PRECOLLEGE COMPUTER CURRICULA: A SYMPOSIUM.....1
2. DATA PROCESSING MANAGEMENT ASSOCIATION.....4
3. COMPUTER SCIENCE FOR SECONDARY SCHOOLS: COURSE CONTENT.....19
4. THE REPORT OF THE ICTM SC^3.....40
5. ASSESSING LIBRARY MICROCOMPUTER NEEDS.....59
6. WORD PROCESSORS: "GHOST WRITER".....62
7. TELECOMMUNICATIONS FOR EDUCATORS - AN INTRODUCTION.....66
8. STRUCTURED BASIC ON THE MACINTOSH.....72
9. ISSUES AND TRENDS.....77
10. COMPUTERS IN BIOLOGICAL EDUCATION.....79
11. BASIC FOR STUDENTS WITH A LIMITED MATHEMATICS BACKGROUND.....82
12. WORD PROCESSING: INTRODUCTION IN THE COMPUTER LAB.....86
13. MICROWORLDS AND MACROWORLDS.....90
14. PROFESSIONAL TRAINING IN CBE: THE VIEW OF ONE UNIVERSITY.....95
15. KINDER KEYS, A KEYBOARDING PROGRAM.....97
16. LASAGNA CODE...AN ALTERNATIVE TO SPAGHETTI.....99
17. SOFTWARE EVALUATION IS A PERSONAL AFFAIR.....100
18. COMPUTER EXPEDITION 1984: WHAT TO DO WHEN THE NOVELTY IS GONE.....103
19. EFFECTS OF MICROCOMPUTER TECHNOLOGY ON YOUNG HANDICAPPED CHILDREN.....107
20. THE ROLE OF THE MICROCOMPUTER AT ILLINOIS STATE UNIVERSITY.....110
21. POLISHING THE APPLE IN VOCATIONAL EDUCATION.....113
22. KEYBOARDING IN THE K-12 CURRICULUM.....116
23. THE MICROCOMPUTER LAB OF THE FUTURE.....118
24. TAKING THE NEXT STEP - SLOWLY!.....122
25. CAI--IS THAT ALL THERE IS?.....127

26. OUTLINING A PRACTICAL FIRST COMPUTER COURSE FOR SPECIAL EDUCATORS.....131
27. COLLEGE CHEMISTRY GRADUATES HAVE POOR COMMUNICATION SKILLS.....134
28. COMPUTER POWER: A PROBLEM SOLVING COURSE FOR ALL LEVELS.....136
29. PASCAL AS A PROGRAMMING LANGUAGE FOR BEGINNERS.....140
30. WORD PROCESSING IS GOOD FOR ELEMENTARY TEACHERS AND STUDENTS.....142
31. KEYBOARDING.....145
32. TEACHERS TALK--THE FAILURE OF COMPUTERS IN EDUCATION?.....148
33. THREE ANCHORS AND COMPUTER-ASSISTED LEARNING.....153
34. INCREASING FEMALE INVOLVEMENT IN THE USE OF COMPUTERS.....161
35. USING NEW TECHNOLOGIES IN EDUCATION: A REDEFINITION.....163
36. CHILD TRACKING SYSTEM.....170
37. COMPUTER LITERACY: STILL AN ISSUE OR SHOULD WE CHANGE ITS NAME.....173
38. IMPROVING INSTRUCTION THROUGH APPROPRIATE SOFTWARE FOR SPECIAL ED.....177
39. TAKING THE BYTE OUT OF COUNSELING.....184
40. SWEETENING YOUR APPLE.....191
41. THE COMPUTER AS A TOOL IN THE JUNIOR HIGH CLASSROOM.....193
42. IMPLEMENTATION OF A K-8 COMPUTER CURRICULUM.....196
43. CIC--COMPUTERS IN THE CURRICULUM.....200
44. EXCELLENCE IN ADMINISTRATION: PLANNING AND PRESENTING.....205
45. ADVANCED TECHNIQUES OF LOGO PROGRAMMING.....209
46. ATHLETIC ADMINISTRATION....SPORT STATISTICS.....217

Appendices

A. INDEX 223
B. VENDOR LIST.....226
C.LIST OF SPEAKERS.....229

FOUR PRECOLLEGE COMPUTER CURRICULA: A SYMPOSIUM

By: Peter Lykos, Illinois Institute of Technology, Chicago, IL

Professional societies that focus on aspects of computer use and development have given attention to curricular needs at the high school level. This symposium will involve four professional society committees who have already produced (either in final form or at an advanced stage of development) curricula for high schools based on the computer, its use and development.

First I point out the existence of the American Federation of Information Processing Societies (AFIPS). AFIPS began with three charter member societies and has grown to an eleven professional society federation. The American Chemical Society, a 135,000 member society with an annual budget of 115 million dollars, has begun the process of negotiating its membership in the federation.

AFIPS conducts an annual National Computer Conference usually held during the summer which draws about 100,000 participants and over 600 exhibitors. The technical sessions span five days in some dozen parallel tracks. The income from the NCC results in several hundred thousand dollars to each of the charter member societies as well as additional income to the noncharter member affiliates. AFIPS maintains a Washington, D.C. office with designated representatives of its constituent societies who are ready whenever some part of the federal government needs technical counsel involving its computer policies, AFIPS can provide assistance very quickly.

AFIPS is made up of the following professional societies:

American Society for Information Science (ASIS)
American Statistical Association (ASA)
Association for Computational Linguistics (ACL)
Association for Computing Machinery (ACM) - Charter Member
Association for Educational Data Systems (AEDS)
Data Processing Management Association (DPMA) - Charter Membership Status Negotiated
Institute of Electrical and Electronic Engineers Computer Society (IEEE-CS) - Charter Member
Instrument Society of America (ISA)
Society for Computer Simulation (SCS) - Charter Member
Society for Industrial and Applied Mathematics (SIAM)
Society for Information Display (SID).

Our keynote speaker, Dr. Sylvia Charp, just completed a two year term as President of AFIPS. Given her background in computer use in the Philadelphia School System it will come as no surprise that while she was AFIPS President she brought into being a project out of the AFIPS Education Committee toward high school curricula and started it out with $50,000 of AFIPS support. That project started with a steering committee (of which I am a member) which in turn spawned a working committee. I was able to add to that working committee an `in the trenches' high school teacher (Margaret King of Oak Park-River Forest High School who, over the past five years, has developed a 27 terminal time-sharing system to support the OP-RF High School Business Program. Ms. King is on sabbatical leave to NIU this year finishing her doctoral thesis).

Shortly after that working committee was formed, the direction of the project switched from consideration of three tracks designed to start students down the roads to computer science or computer engineering or information systems to a curriculum which would develop for the student the entire range of computer impact on society. The working committee is meeting in Washington, D.C. in mid February, 1985, and has a working draft of its product with the following principal headings:

I. Origins and Roles of Information (3-5 weeks)
   1. Information environment
   2. Historical development of information explosion
   3. Roles that information plays in society

II. How Information is Processed (18-24 weeks)
   4. Processed by humans
   5. Processed by computers
   6. Human-Computer interaction

III. How Information is Used (7-9 weeks)
   7. Model building
   8. Technology and tools
   9. Information resources

IV. How Information Affects Us (6-8 weeks)
   10. Careers and education
   11. Issues
   12. Future

Clearly, that represents a broad and all-encompassing outline. A thorough analysis of the already in-place programs at a specific high school would need to be made in order to determine just what portion of that outline should be taught in what part of the curriculum - and by which teachers.

The other professional societies which have prepared curricula will be heard from in the talks subsequent to this one as follows:

Dr. Willard Daggett, of The NY State Dept. of Education, will describe the genesis and the setting whereby the Data Processing Managers Association (DPMA) 9-12 computer information systems curriculum evolved.

Prof. Jean Rogers, of the University of Texas at Austin, will describe the high school curriculum for the discipline of computer science itself. The curriculum she will describe bears the endorsement of both the Association of Computer Machinery (ACM) and the Institute of Electrical and Electronic Engineers (IEEE-CS).

The Illinois Council of Teachers of Mathematics (ICTM) formed a committee charged with producing a two-year curriculum suitable for secondary schools of Illinois. The 10 member committee, co-chaired by Lowell Carmony, of Lake Forest College, and Don Enns, of Highland Park High School, completed its report in November, 1984, and a panel of four members of that committee will present the report and respond to questions.

Other professional societies where the computer is having a significant impact are also pondering the places in the curriculum at all levels of precollege and college education where the computer-based enhancements to the discipline ought to be incorporated. Perhaps this will become a regular feature of MICRO-IDEAS conferences!

DATA PROCESSING MANAGEMENT ASSOCIATION

SECONDARY CURRICULUM

ON INFORMATION TECHNOLOGY AND COMPUTER INFORMATION SYSTEMS

First Edition, October, 1984

A service of Data Processing Management Association, 505 Busse Highway, Park Ridge, IL, 60068, (312) 825-8124

Permission to reproduce in the proceedings of The Role In Education 5 granted by: Hildegard A. Klemm, Manager of Educational Services, DPMA

Copyright 1984. Data Processing Management Association, 505 Busse Highway, Park Ridge, IL 60068 USA. ALL RIGHTS RESERVED. No part of this book may be reproduced by any means, electronic, mechanical, photocopying, recording or otherwise, nor transmitted, nor translated into a machine language for storage and/or retrieval, without the written permission of the DPMA except for brief excerpts to be used in published reviews.

INTRODUCTION

This curriculum, sponsored and supported by the Data Processing Management Association (DPMA), responds to needs of secondary students for:

- Computer literacy
- Marketable, computer-related skills in occupational education programs
- Increased knowledge about computer information systems to meet more stringent college-entrance requirements.

The curriculum outlined on succeeding pages provides a basis for meeting all these needs. Thus, this curriculum relates specifically to the evolving needs of students with declared interests or career potentials involving control over or active use of computers and computer information systems. Included are the areas of business management, business administration, and information systems. Content of this curriculum will also be valuable for students going into a number of other major disciplines, including music, law, journalism, medicine, and education.

Note, however, that this curriculum does **NOT** cover some specific areas of computer education. These areas, and the reasons for not including them in this curriculum, are given below:

- Computer science, the discipline that emphasizes systems-level configuration and design of software, is not covered because specialists in this area have little involvement with business applications, the main focus of this curriculum.
- Computer engineering is the discipline specializing in the design and fabrication of computer hardware. In a world of general-purpose computers, in-depth technical knowledge of equipment is not essential to the appreciation of business problem solving, toward which this curriculum is targeted.

DPMA, a leading professional organization in the computer field, has some 47,000 members whose careers are associated directly with business data processing and information systems. DPMA, through its Education Foundation, has already supported the development and implementation of a college-level Undergraduate Curriculum in Computer Information Systems. Work at the college level served to identify needs and opportunities for business-related computer education at the high school level. This curriculum is an outgrowth of efforts initiated and supported by DPMA.

The present curriculum evolved over a period of two years as a result of constructive, concentrated efforts by a group of educators who donated their time and experience to bring this curriculum into being. Contributors to the successful development of this curriculum document have included:

Dr. Willard Daggett, New York State Department of Education, Albany, NY
Dr. William Pasewark, Texas Tech University, Lubbock, TX
Dr. Gary M. Lloyd, Utah Office of Education, Salt Lake City, UT
Dr. Wilma Jean Alexander, Illinois State University, Normal, IL
Dr. Donald B. Medley, California State Polytechnic University, Pomona, CA
Dr. David R. Adams, Northern Kentucky University, Highland Heights, KY
Mr. Kamiran Badrkhan, Los Angeles Trade Technical College, Los Angeles, CA
Dr. James M. Doyle, Sweetwater Union High School District, Chula Vista, CA
Dr. Robert Poland, Michigan State University, East Lansing, MI
Dr. Victor Van Hook, Oklahoma State University, Stillwater, OK

ORGANIZATION

Courses outlined and/or identified in the accompanying curriculum are structured into four levels:

- **LEVEL ONE** includes a single offering, an introductory, one-semester course designed to bring students to a level of computer literacy that is universally acceptable under state and local curriculum guidelines and, as an option, to provide initial, hands-on experience in use of typical microcomputer applications. The Level One course can be administered to students in grades seven through nine.

- **LEVEL TWO** provides students with a career-related indoctrination to computers and their applications. The course identified for this level can be administered in either one or two semesters and provides further background in the need for and development of computer information systems. This course also establishes an understanding of computer programming and introduces students to exercises leading to development of their first business-type programs. The Level Two course is intended for students in grades eight through 10.

- **LEVEL THREE** consists of a single, one-semester offering that should carry the status of an honors course. This program covers the same content currently incorporated in many freshman-level introductory courses required at colleges. There is a widespread movement at the college level to shift this content from requirement to prerequisite. As this happens, there should be great attraction for enrollment in this course among college-bound students, particularly those interested in business or related fields. In addition, this course will be appropriate for occupation education students intending to seek careers in the computer information systems field. The Level Three course is suggested as a twelfth-grade elective. As an option, this course could be offered to eleventh-grade students who have demonstrated high levels of competence in completing coursework at levels one and two.

- **LEVEL FOUR** courses impart job competency skills. These offerings are suitable for both professional advancement and for the building of entry-level job skills. Level Four courses are single-semester, practical training modules intended for grades 11 and 12. Content for individual courses in this series will be outlined in some depth in future editions of this curriculum document.

IMPLEMENTATION

Provisions are being made to assure availability of student materials to support all course offerings. It is anticipated that the first texts and supplementary materials will be be available for use in the Fall semester of 1985. Additional materials will be produced on an expedited schedule. It is expected that all of the core courses and a number of the occupational skills courses will be supported by the fall of 1987.

Arrangements are being made to provide extensive, in-depth teacher training in direct support of this curriculum and its pedagogical principles. The target for this effort is to provide 30 hours of continuing professional education to interested, motivated teachers.

Other teacher-assistance programs, including a newsletter and reference hot-line, are also anticipated.

COURSE OFFERINGS

The remainder of this document is devoted to presentations of titles, descriptions, and outlines of the courses that will make up the DPMA Secondary Curriculum on Information Technology.

Course Title: COMPUTERS AND INFORMATION TECHNOLOGY

Curriculum Level: Level One

Grade Level: Seven through Ten (completed in one semester)

Objectives:

- Students will become computer literate.
- Students will be able to identify and describe the basic parts of the information processing cycle: input, processing, output, and storage.
- Students will be able to identify the working parts of computer equipment systems, including processors (arithmetic logic unit and control unit), main memory, terminals, printers, disk drives, tape drives, plotters, audio response devices, and others.
- Students will be familiar with the operation of personal computers (microcomputers). If personal computer equipment is available, students will, at the discretion of the teacher, be able to perform simple word and data processing tasks on personal computers.
- Students will be able to discuss the need for an orderly pattern in the development of computer systems, including an explanation that all computer systems must be designed to meet user needs.
- Students will be able to identify steps in a process for program development and will be able to describe the purpose of each step.
- Students will be able to explain the role of systems software, including operating systems.
- Students will be able to describe the purpose and functions of programming language software.
- Students will recognize and be able to describe functions of a series of commands in the BASIC programming language. As an option to be implemented at the discretion of the teacher, the students may be able to enter and execute a simple BASIC program for which coding is provided.
- Students will be able to present comparative descriptions of conventional document preparation techniques and word processing methods.
- Students will be able to describe procedures associated with word processing on personal computers. As an option, students may be encouraged to complete simple word processing exercises on microcomputers.

- On being presented with key job titles in the computer field, students will be able to identify the roles and responsibilities of persons who hold those positions.

- Students will be able to understand and discuss a series of typical computer applications, including transaction processing, financial reporting, computer-aided design, computer-aided manufacturing, robotics, and computer-assisted instruction.

Description:

This program is designed as a first course in computer capabilities and information technology.

There are no prerequisites. The course should be equally appropriate for students following either college-bound or occupational curricula.

The course should equal or surpass standards for any state with a computer literacy requirement.

It should be possible to administer the curriculum, outlined below, in a single semester if instruction is given for one full period daily for at least 16 weeks.

It is recommended strongly that, if at all possible, students be given some opportunity for hands-on use of computers in conjunction with this course. This exposure need not be extensive if computing equipment is in short supply. However, it is stressed that as little as three or four hours of first-hand experience will serve to overcome inhibitions and will represent an important step toward the education of future computer users. (Self-study modules in beginning keyboarding skills can be used as part of the program to meet this objective. At the time hands-on computer experience begins, all students should be expected to display elementary keyboarding proficiency.)

Teachers assigned to this course will not require advanced skills in data or information processing. However, the teacher assigned to this course should have a level of knowledge that makes for comfort in explaining and responding to questions associated with the curriculum outlined below.

Content Outline:

I. Information as a Basic Human Need
   A. Define data and information
   B. Role of computers
   C. Case situation: Illustrate point that everyone's life is affected by computers

II. The Information Society
   A. Historical need for and evolution of data and information processing
   B. Birth of modern data processing
      1. Payroll deduction-social security
      2. World War II-massive personnel groupings; new technologies

C. Evolution of the computer
   1. Punched card equipment
   2. Introduction of computers--rapid growth
   3. The microcomputer (personal computer) phenomenon
   4. The growing computer population
D. Issues and problems associated with computers
   1. Privacy
   2. Security
   3. Need to deal with offenders

III. The Information Processing Cycle
A. Input
B. Processing
C. Output
D. Storage

IV. Processing Information on Computers
A. Data and information resources
   1. Records
   2. Files
   3. Data media
B. Input devices and functions
   1. Manual data entry
   2. Batch entry
   3. Direct entry
   4. Data capture as part of source transactions
   5. Audio and other
C. The processor
   1. CPU: ALU and control unit
   2. Main memory
D. Output devices
   1. Printers
   2. Plotters
   3. Audio
   4. COM and other
E. Storage
   1. Tape
   2. Disk
   3. Diskette
   4. Cassette

V. Computer software
A. System software
   1. Operating systems
   2. Utilities
B. Programming languages
C. Application programs
D. Text development and editing

VI. Document and text processing
A. The typewriter
B. Mainframe text processors
C. Word processors
D. Word processing on personal computers

VII. Computer applications (Data oriented)
- A. Transaction processing
- B. Financial reporting
- C. File and data base building and maintenance
- D. Analysis
- E. Forecasting
- F. CAD/CAM and robotics

VIII. Information system and program development
- A. Need for step-by-step processes
- B. Steps in system development
- C. Steps in program development
  1. Relationships between system and program development

Course Title: COMPUTER INFORMATION SYSTEMS AND PROGRAMS

Curriculum Level: Level Two

Grade Level: Nine through Eleven

Objectives:

- Students will review and reinforce their command of basic principles of data and information processing.
- Students will be able to explain the need for a planned, orderly process for information system development.
- Students will be able to identify data flow diagrams, structure charts, and flowcharts and explain the roles of these tools in information system development.
- Students will be able to describe the role of program development in relation to overall development of information systems.
- Students will be able to identify and describe the basic steps in the process of program development, including problem analysis, program design, program specification, program coding, and debugging and testing.
- Students will be able to describe the role and value of breaking overall programming jobs into a series of small modules.
- Students will be able to present and describe the three basic functions that form the hierarchy of program organization.
- Students will be able to prepare simple structure charts that identify and relate modules of rudimentary programs.
- Students will be able to develop sets of pseudocode statements describing the processing steps necessary to implement programs represented in structure charts.

- Students will be able to draw simplified system flowcharts to reflect the processing sequences of individual program modules.

- Students will be able to use a simplified subset of BASIC or other appropriate programming language to code programs for which they have developed structure charts and pseudocode.

- Students will be able to create simple, small files in computer memory for use in testing programs they code.

- Students will be able to debug and test simple programs, including programs with multiple modules.

Description:

For full benefit from this course, students should have a prerequisite level of computer literacy. This can be attained through completion of the course entitled COMPUTERS AND INFORMATION TECHNOLOGY or through equivalent studies and readings.

Teachers should be aware that the emphasis in this course is on information processing. Students who have completed previous work in courses stressing mathematical uses of computers or computer sciences may need special reviews in the principles of information processing, particularly the concepts dealing with design of information systems to meet user needs.

This course will lead students through requirements for controlled development of information systems. Alternate techniques for achieving this controlled development will be described. These presentations will establish a context for students to understand that computer programming is also a step-by-step process. Students will learn to appreciate the fact that there is a lot more to computer programming than the simple writing of code in a programming language. Students will understand that the writing of code can be reduced to a simple, clerical-type function if programs are designed and planned effectively.

Students will be exposed to and given an opportunity to master proven tools for program development, including preparation of structure charts that break overall programming jobs into a series of manageable-sized modules. Students will also learn to write pseudocode, sets of structured English statements that describe the steps needed to implement a solution of an information processing problem on a computer.

Students should learn a series of commands that can be used for coding problems in BASIC or another appropriate programming language. They should be able to develop programs using these commands and to enter their programs into a microcomputer (personal computer). They should also be able to follow through to debug and test their programs. Successful students should be able to deliver specified outputs when test data are processed under control of the programs they have written.

This course can be taught as a one-semester survey or can be carried through two semesters.

Content Outline:

I. The information system--a review
   A. The information processing cycle
   B. Basic information system functions
   C. Role of computers
   D. The microcomputer as an information processor
      1. Hardware
      2. System software
      3. Application development and processing

II. Information system development
   A. Life cycle approach
   B. Computers as system development tools--prototyping
   C. Systems analysis through the data flow approach
   D. Systems analysis through the data structures approach
   E. Systems design
   F. Developing system specifications
   G. The role and place of programming within information system development

III. Principles of computer application programming--a review
   A. Programming languages
   B. The computer application
   C. Relationships of programming languages, applications, and system software
   D. Application development on personal computers

IV. A process for program development
   A. Problem analysis
   B. Program design
   C. Program specification
   D. Program coding
   E. Program debugging and testing

V. Programming languages (BASIC, Pascal, or COBOL)
   A. Practical, working subset of programming language commands
   B. Program development example
   C. Practice exercises in program development

VI. Application program development--a case study
   A. Case situation
   B. Step-by-step solution

VII. Program development--practice assignments
   A. Case situation presentations
   B. Instructions for program development, debugging, and testing
   C. Solutions provided in teacher's manual

Course Title: COMPUTERS IN MODERN LIFE

Curriculum Level: Level Three

Grade Level: Twelve

Objectives:

- Students will be able to discuss the impact and role of computers as they affect modern life.
- Students will be able to explain that the development of data processing and computing equipment met developing demands--that computers have been a response to demands rather than a driving force creating change for its own sake.
- Students will be able to identify and describe the basic functions of all data processing systems--functions that are present regardless of whether computers are used or whether processing is done manually, and regardless of the size or configuration of computer to be used.
- Students will be able to explain the need for and basic structural components of computer information systems (CIS) departments in large organizations.
- Students will be able to describe the situations and requirements that lead to installation and implementation of large, mainframe computer systems.
- Students will be able to describe the organization structures and job opportunities in CIS departments of large organizations.
- Students will be familiar with the functions and operations of the major equipment components of mainframe computer systems, including processors, consoles, tape drives, disk drives, printers,other output devices, data input sections, data control sections, programming groups, and systems development sections.
- Students will be familiar with project structures and other approaches used to develop comparatively large computer information systems.
- Students will be able to describe the roles and functions of data processing, management information, and decision support systems.
- Students will be able to describe an organized system for program development and to implement program development procedures, including creation of structure charts, pseudocode, system flowcharts, file structures, and test data.

Description:

This course is intended as a twelfth-grade honors program. Enrollees can include college-bound students and occupational education students who have displayed strength in computer-related skills and have expressed an interest in a career in the CIS field.

One of the targets of modern CIS education at the college level is to have students come out of high school with a level of knowledge equivalent to the instructional content of present-day freshman introductory courses. These introductory courses are, today, typical requirements for business, accounting, and CIS majors. As computer instruction increases in secondary schools, it is anticipated that this basic course will be reduced to the level of prerequisite--rather than being offered as a credit course. This course will satisfy the anticipated prerequisite requirement. For the near term, students who complete this course can probably anticipate advanced placement when they enter college.

This course marks a break from earlier courses in the DPMA secondary curriculum in that it deals principally with large, mainframe computer systems and careers related to such systems. The topics of systems and program development are treated in greater depth in this course than in earlier offerings within this curriculum. Implementation of this course will benefit from use of guest lecturers and from field visits to appropriate, large computer installations.

An extensive study supplement and accompanying instructor's manual will make it feasible for high school teachers to deal effectively with the content of this course.

Content Outline:

I. Computers in modern society--role and extent

II. The need to process data and develop information--evolution of data processing and computer information systems

III. The data processing system--basic functions for manual, mechanized, or computerized systems

IV. Computer-based systems

V. Computer processors

VI. Computer system peripherals and terminals

VII. Programming principles

VIII. Types of computer information systems

IX. Computer applications in business

X. Computer information systems development

XI. Computer system configurations--microcomputers to mainframes

XII. Computer software and programming languages

XIII. The CIS function in business and governmental organizations

VIV. Information delivery

XV. Information sources

XVI. Files and databases

XVII. Computer operating systems

XVIII. Distributed data processing--including networking

XIX. Careers in information processing

Course Title: OCCUPATIONAL EDUCATION OFFERINGS

Curriculum Level: Level Four

Grade Level: Eleven and twelve

Description:
Courses at this level will be aimed at developing specific job competencies and/or advanced skills. The listing below represents only a preliminary set of course titles, grouped for similarity of subject content. It is anticipated that this list will be expanded as experience builds and as developments in the computer information systems field warrant.

Course Titles (by group):

Business Computer Programming

- Computer Program Design and Development
- Advanced BASIC Programming
- Programming in Pascal
- Introduction to COBOL Programming

Business Computer Operations

- Computer Operations and Services
- Computer Data Entry
- Computer Architecture--Hardware and Software
- Computer System Documentation

Microcomputers in Business

- Microcomputers for Business Applications
- Electronic Records Management
- Introduction to Database Concepts
- Preparing Financial Reports on Microcomputers
- Using Spreadsheet and Analysis Programs

Keyboarding

- Basic Keyboarding Skills for Computer Users
- Executive Keyboarding

Note: These courses will be expanded with full outlines in subsequent editions of the DPMA Secondary Curriculum.

COMPUTER SCIENCE FOR SECONDARY SCHOOLS: COURSE CONTENT

In 1983, a task group working under the ACM Education Board's Elementary and Secondary Schools Subcommittee drafted recommendations for courses in computer science at the secondary school level. The task group was made up of secondary school teachers, public school computer curriculum specialists, university computer science teachers and computer science education specialists. The recommendations have been reviewed by over 100 individuals involved in computer science education and presented and discussed at several conferences. As a result of suggestions by reviewers and respondents, refinements have been made. The final report of the recommendations presented in this document, has been accepted by both the Association for Computing Machinery's Education Board and the Education Committee of the IEEE-Computer Society.

COMPUTER SCIENCE IN SECONDARY SCHOOLS

The Committee report focuses specifically on computer science in the secondary school. Computers and computing are topics of discussion in many curriculum areas in secondary school. Computers can be found in classrooms, laboratories, and resource centers. Computer use is being integrated in a wide variety of courses and instruction. This committee supports the integration of the computer into the school curricula, using computing wherever it is appropriate.

The four courses described in this document, however, have computing as their primary content. The courses are proposed as models for secondary schools to use in introducing the academic discipline of computer science. They are not vocational, nor necessarily pre-professional, but rather part of the secondary school student's general education.

The Committee recognizes that changes occurring in both computer science and education will create a need for changes in courses about computing. In fact, future evolution may make some of these courses unnecessary. Even at this time, some schools will find that students at the secondary level will have previous knowledge about course topics; in these cases, teachers should modify the content to make the courses more appropriate. The courses proposed here are quite rich, and virtually all topics suggested can be studied to great depth. In adapting these courses, however, their breadth should not be narrowed. The four courses are:

1) Introduction to Computer Science I (a full year course)
2) Introduction to Computer Science II (a full year course)
3) Introduction to a High-Level Computer Language (a half-year course)
4) Applications and Implications of Computers (a half-year course)

Courses 1, 3 and 4 have been designed assuming no prerequisite courses and are not interdependent. Course 2 assumes Course 1.

INTRODUCTION TO COMPUTER SCIENCE

Courses 1 and 2 are designed for students with a strong interest in computer science. Course 1 can serve as a single introductory course for some students and also act as a prerequisite for Course 2. At the end of two years study, students should be prepared to be placed in second year computer science classes in post-secondary educational institutions, either directly or through the Advanced Placement Exam available through the College Entrance Examination Board.

Courses 1 and 2 have been designed to coordinate with several other previously described curricula (see references). The structure of Courses 1 and 2 differ from other curricula in two important ways.

1) The content of a typical university first year sequence has been divided here to be taught over two years time. This reflects the typical relationship of high school to college year courses, where standard high school courses are presented at a slower pace. For example, two years of a foreign language in high school is often considered equivalent to one year university level.

2) The means of division of the content between Course 1 and 2 has been a horizontal rather than vertical cut. Instead of the first course containing simply the first half sequentially of the material, it has been designed to cover the breadth of the material, with only moderate depth. The second course then addresses the same topics, but to a much greater depth. Thus, the student who takes the first course but does not continue to the second will study a coherent whole instead of a fraction. Meanwhile, the student who chooses to continue to the second course has a solid footing for more extensive work. This is a typical model in secondary schools where, for example, many students take Biology and a few study topics to greater depth in Advanced Biology.

Courses 1 and 2 will require a significant amount of equipment for each student enrolled. Computers, either microcomputers or a timesharing system with convenient storage for files (disks) and high-speed printers are necessary.

GENERAL EDUCATION COURSES

Courses 3 and 4 are designed to be of general interest to any student at the secondary level. Course 3 is a course about programming. It is a general introduction to the process of writing programs in a high-level computer language. It requires extensive hardware facilities, as students will need to practice using whichever computer language is chosen for the instruction

Course 4 includes information about the ways computers are used and about the impact of computer use on people's lives. Because students in this course will be using a computer to learn about applications, teaching this course requires extensive software in addition to hardware. While some examples of applications programs could be demonstrated by the teacher to the class, in general students should be running these computer programs themselves.

A school district that decides to require a single, half-year course of all students could choose either Course 3 or Course 4. If a district requires a full year of study of computing for all students, Courses 3 and 4 would be a reasonable requirement. The choice should depend on the school district's K-12 computer education plan. If only one course will be required, the district should establish which type of computer coursework has priority: knowledge of applications and implications, or programming. In general, the choices should be made as part of a total educational plan and applied to all students, not left as a student's option.

TEACHERS

Teachers of all four of these courses should be qualified in the content area and preferably certified to teach about computing and computer science. The Committee recognizes that adequately trained teachers are in short supply compared to the number needed to teach these computer science courses. Preparing these teachers should be a critical, immediate goal for teacher training institutions. Recommendations for teacher training and certification is the topic of a concurrent report prepared by a second task group working through the ACM's Elementary and Secondary Schools Committee.

Textbooks, support materials and computer-based instruction modules appropriate for these courses will help in providing sound instruction; some such materials now exist or are being developed. However, the importance of knowledgeable, dedicated teachers to the success of these computer science courses should be emphasized. The discipline of computer science has been evolving over several years and will continue to change. Courses in computer science will similarly continue to change. New concepts and machine capabilities will continue to evolve and become an appropriate part of these courses; current course topics will become so commonplace that they can be dropped from instruction at this level. Well-trained teachers are the cornerstone of effective instruction in such a fluid environment.

FACILITIES

Facilities needed for these four courses are extensive. In each course, both hardware and software will be required. Students will frequently need time for computer use outside formal class hours, so sufficient machines and flexible access must be provided. The actual time per student will vary depending on the course. Courses 1 and 2 will require more per student than Courses 3 or 4. The typical school, however, would have many more students in Course 3 or 4, so the total need for equipment per course would be approximately the same.

School districts that do not have sufficient funding to provide all the necessary equipment for teaching the complete set of these courses should establish a priority among them, realizing that the courses are designed to serve different audiences. It is better to teach only one or two courses using adequate facilities than to teach all four without sufficient machine access and software availability.

PROGRAMMING LANGUAGES

Courses 1, 2, and 3 require that students learn to write computer programs as part of the course work. The choice of language for a particular course will

depend on a number of considerations.

The specific language that is chosen for students to learn during Course 1 should be one that encourages and facilitates modular, structured programming. While students can be taught good programming practices whatever language they are learning, some languages are more supportive of these accepted practices. Structured BASIC and Pascal are readily available vehicles in the educational environment; either is appropriate for this course.

The programming language used in Course 2 should be any high-level, structured language. Pascal is currently the most common language used in courses such as the one described here. If students are taking this course to prepare for the College Entrance Examination Board's Advanced Placement exam, then Standard Pascal would be preferred.

Course 3 allows more flexibility in language choice, as it is not a pre-professional course. Its main purpose is to introduce the student to the idea of computer programming rather than to the academic discipline of computer science. Therefore, an important consideration in language choice is the ease with which a beginner can get started and enjoy success. Both BASIC and LOGO allow beginners to write an operational program in a short time. By the same token, an interpreted language typically makes early success by the novice more attainable than does a compiled language, because the programming environment is less complex.

The demands of a structured language, such as Pascal, will require that the teacher and learner spend more time on rules of syntax, than for unstructured languages. Some computer scientists, however, feel that learning an unstructured language permanently affects an individual's ability to learn a structured language; these computer scientists recommend that no unstructured languages be taught.

In addition to LOGO, BASIC, structured BASIC and Pascal, there are a number of other languages currently in or entering educational computing. An individual school district may decide to choose one of these languages for Course 1, 2, or 3, but should do so with careful consideration of the impact of the choice.

COURSE 1
INTRODUCTION TO COMPUTER SCIENCE I

Prerequisite
None

Goals

This course includes the basic ideas of structured programming and top-down design. Students should learn to design (structured) algorithms to solve (programming) problems. Problems should be broken down through a series of stepwise refinements into logical modules. Although a specific language must be used, the coverage of algorithms and design techniques should be general enough so that the student can use this course as a foundation for structured problem solving in any language. This course is an introduction to the academic discipline of computer science or a first step in professional training, but not a vocational course.

To cover all of the course topics to the level of detail described will require from 50 to 75 class hours of instruction time in addition to tutoring sessions (up to 18 hours) and laboratory time. The laboratory time required to do a basic set of assignments varies depending on the pace of the course and the background of the students. However, 2 hours per week per student alone at a computer is a minimum. This time is critical to the success of the course since students also learn from coding, testing and debugging programs. In some cases, students may be working together on group programming projects. Group projects have educational value particular to them. Nonetheless, sufficient computer time must be provided for each individual.

Objectives

The student should learn general problem solving techniques and be able to:
- clearly define problems,
- sub-divide problems into logical modules,
- design structured solutions to problems (algorithms),
- code algorithms into a specific language,
- hand trace algorithms and code, and
- test and debug programs.

The student should understand the uses and limitations of a high level language with regard to:
- the available data structures, and
- the available control structures.

The student should be able to implement solutions in a specific high level language for the major types of programming problems.

The student should be able to anticipate, identify, isolate and correct errors.

Topics

The following topics are grouped to show the scope and depth of the course. They should not be covered in the order that is shown below. Instead the topics should be intervoven as necessary to develop the ideas inherent to structured programming. The actual sequences presented by most structured programming textbooks are very similar. The system is usually discussed first, then the basic syntax and semantics are introduced. This is followed by a thorough treatment of the control structures and then data structures. Program correctness and design are discussed throughout.

Because of the practical limits (time available and maturity of students in secondary school) all the topics in the following outline cannot be covered in great depth in a one-year course. Specifically, the applications of algorithms listed in VII.C involve a wide range of significant material. When choosing content for the course, the teacher should include all topics briefly rather than eliminate some of them. This course is intended to cover the breadth of an introduction to computer science while a more in-depth study of these same topics is left to the following course.

I. Programming Environment (system hardware & organization): (5%)
- A. Editing and creating files.
- B. Compiling, linking, running programs.
- C. Inputing and outputing of data.
- D. Tracing an actual program from editing to final run.
  - 1) the files,
  - 2) the code,
  - 3) the operating system.

II. Formalization of the Language: (5%)
- A. Syntax vs. semantics of the language.
- B. Organization of the language.
  - 1) the reserved words, symbols, predefined identifiers, etc.
  - 2) combining words and symbols to form logical structures.
  - 3) program format (syntax).

III. Available Control Structures: (20%)
- A. Selection.
- B. Repetition.
- C. Subprograms.

IV. Available Data Structures: (20%)
- A. Simple variables that store
  - characters,
  - numbers (integers, reals, etc.),
  - Booleans,
  - user defined ordinal types.
- B. Arrays: one-dimensional
  - others
    - multi-dimensional,
    - arrays of complex types,
    - packed arrays (strings).
- C. Records.
  - simple,
  - hierarchical,
  - (variant only if time permits).
- D. Sets. Applications.
- E. Files of characters (TEXT),
  - other types.
- F. Dynamic data structures using pointers.

V. Operators: (10%)
- A. Built in operators for simple data types:
  - integers,
  - reals,
  - characters,
  - Booleans,
- B. Built in operators for complex data types:
  - packed arrays
  - files,
  - records.

C. Built in Functions, Procedures:
Abs, Cos, Trunc, etc.
Reset, Rewrite, etc.

VI. Program correctness: (10%)
A. Anticipating errors.
B. Tracing by hand & computer.
C. Testing through running a program.
D. Debugging.
E. Verification.

VII. Algorithms: (30%)
A. Design methodology for problem solving
1) problem definition & specifications.
2) stepwise refinement into logical parts.
B. Development of algorithms
1) choosing data structures to match the problem.
2) choosing control structures to match the problem.
3) Designing modules
procedures,
functions,
parameters,
scope.
4) style, clarity, structure.
C. Applications of a high level language (such as Pascal) to representative and classical problems.
Reading and writing data formatted and unformatted,
Using counters, flags, cumulative results
(high, low, total, etc.)
Filling, copying, processing arrays,
Record manipulation by index and field,
Writing, filling,m comparing strings (packed arrays),
Sorting by
bubble,
selection,
insertion.
Searching by
index,
contents.
Merging of
arrays,
files.
Recursion vs. iteration for
calculations,
searching.
Processing dynamic structures of
linked lists,
stacks, queue, tree.

COURSE 2
INTRODUCTION TO COMPUTER SCIENCE II

Prerequisite
Introduction to Computer Science I

Goals

This course is designed to build and expand upon the concepts students have learned in INTRODUCTION TO COMPUTER SCIENCE I. In this course, students would typically meet with the teacher in a regular, daily class for an academic year, and spend an additional 45 to 60 minutes per day doing independent work with the computer.

Practice with programming is an important component of this course. Students will be writing more significant programs than they did in the previous course. They should be given assignments to illustrate each major concept discussed in class. Students should also do several 3 to 4 week projects, some of which are group projects, and a term project. Coordination of assignments with specific application areas (as given in V. below) is strongly recommended.

While the students are developing their programming ability through assignments, in-class time can be spent discussing and experiencing computer application tools (e.g. graphic input/output devices) and available software and systems (e.g. an information utility).

Objectives

The student will
- become more adept at problem solving techniques.
- improve in ability to code in a structured language.
- learn the elementary algorithms and data structures of computer science.
- be exposed to computer application tools.
- become familiar with major hardware and software components of a computer system.

Topics

The topics described below are grouped conceptually rather than in the order they would be studied. In several sections, topics are marked with (*). These topics are of low priority in the course content, but may be of interest to particular students or teachers.

I. Review (5%)
   - A. Problem solving strategies
   - B. Structured programming methods
   - C. Computer organization overview

II. Algorithms
   A. Searching (10%)
      1. Techniques
         a) linear
         b) binary
         c) hashing
      2. Efficiency of algorithms
         a) means of measurement
         b) comparison of efficiency
         c) improvement of efficiency
      3. Applications
   B. Sorting (15%)
      1. Techniques
         a) exchange
         b) shell
         c) quick
         d) insertion
         e) selection
         f) merge
         g) radix
      2. Efficiency of algorithms
         a) means of measurement
         b) comparison of efficiency
         c) improvement of efficiency
      3. Applications
   C. Recursion & iteration (10%)
      1. Conceptual differences
      2. Transformation between
      3. Comparison of clarity and efficiency
      4. Verification
   D. Numerical Methods (*)
      1. Techniques
      2. Application areas in mathematics and science
      3. Error analysis

III. Data Structures
   A. Variables (5%)
      1. Representation
      2. Typing
      3. Scope
      4. Parameters
   B. Static data structures (10%)
      1. Structures
         a) array
         b) record
         c) file
      2. Operations
         a) add
         b) delete
         c) search
         d) sort
         e) change
         f) merge
      3. Special vocabulary

4. Choosing a structure
   a) physical vs. logical
   b) advantages & disadvantages
5. Applications

C. Dynamic data structures (15%)
1. Structures
   a) list
   b) queue
   c) stack
   d) tree
   e) file
2. Operations
   a) add
   b) delete
   c) search
   d) sort
   e) change
   f) merge
3. Special vocabulary
4. Choosing a structure
   a) physical vs. logical
   b) advantages & disadvantages
5. Applications

D. Graph theory (*)
1. Concepts and terminology
2. Applications

IV. Computer Organization (10%)

A. Hardware
1. Major components of a system and their functions
2. Memory
3. Processors
4. Peripheral devices

B. Software
1. Operating system
2. Disk operating system
3. Processing
   a) Multiprocessing
   b) Multiprogramming
   c) Time share vs. batch
4. Number systems
5. Low-level languages

C. Networks (*)
1. Major components
2. Major national networks
3. Bulletin boards
4. Applications to business, industry, education

V. Application Areas (20%)

A. Graphics
1. Components of a system
2. Special input devices
3. Special output devices

4. Applications to business and industry
5. Use of a commercial software package
6. Major assignment on this topic

B. Simulation & modeling
1. Characteristics of a system; special languages
2. Quality of assumptions to the process
3. Applications to business and industry
4. Use of several major software packages
5. Major assignment on this topic

C. Information retrieval & storage
1. Types of systems
a) database
b) bibliographic
2. Methods in searching a retrieval system
a) query language
b) Boolean algebra
3. Use a commercial information system
4. Fundamental techniques in storing and retrieving data
5. Major assignment on this topic

D. Telecommunications
1. Definitions and special vocabulary
2. Major components of a system
3. Major suppliers of services
4. Use of services delivered by individuals/organizations
5. Use such a system
6. Applications

COURSE 3
INTRODUCTION TO A HIGH-LEVEL PROGRAMMING LANGUAGE

Prerequisite
None

This section describes a general form from which individual courses could be derived (e.g. "Introduction to LOGO"). The specifics of the individual course will depend on the particular programming language chosen.

Even though the language being studied will have a predominant role in this course, students should realize that knowing programming syntax is not an end in itself, but rather is a means for applying computing to problem solving. Teachers should encourage students to approach program writing in a logical, organized manner.

Goals

The purpose of this course is to provide students with awareness of the reason for a computer language, the role a program plays in computer use, and the ability of the programmer to control the computer. This course is neither vocational nor a prerequisite to any other course described in this document. This course may best be understood as a parallel to an introductory course in music, art or poetry. In a course of this sort, it is presumed the student will be able to understand more than he or she can produce. In a beginning poetry class, students will read and consider the structure of poems when they

could not write an equally good poem. In the same way, students in this class can read and study programs more complex than they are able to write. Student production of programs is not the purpose of this course. The goal is that students will realize the role that programs, and therefore programmers, play in computing.

Topics

Rather than a description of a course, a list of skills for students to develop is given. There are several reasons for this format. The language independence of the stated skills allows flexibility for the school district, the teacher and the student to choose the specific language for the course. Also, many schools provide students with courses in programming languages earlier than secondary school, so students may show that they have these skills at some time prior to the secondary level. Schools also may choose to award credit or waive requirements for students who have gained the skills without taking a formal course. In these situations, a list of skills for the student to demonstrate is more useful than a course description.

The skills specified are within the capability of every secondary school student. Students who take Course 1 should learn the Course 3 skills in the process of their Course 1 work. If a school district decides to require Course 3 of all students, completion of Course 1 should be allowed as an alternative for students with a strong interest in computer science.

Objectives

By the end of this course, the student will

be able to initiate and terminate interaction with a computer.

be able to load and execute an existing program.

be able to enter and edit a program.

be able to recognize characteristics of a well-written program.

be able to identify some common programming errors.

be able to write a correct program that will

accept data from an interactive input device.
display results (numeric, lexical or graphic) on an interactive output device.
assign values to simple variables.
use an iterative structure controlled by a Boolean condition.
use a nested conditional structure.
use an indexed count loop to find an accumulated total.

COURSE 4
APPLICATIONS AND IMPLICATION OF COMPUTERS

Prerequisite
None

The approach to this course is two-fold:

First, the course is oriented toward the student learning to use the computer. The course is intended to provide skills in computer applications that will have powerful meaning in the student's life. In view of this, the course must be success oriented. That is, the student must succeed in using the applications as problem solving tools integrated in their life.

Second, the course is oriented toward the student understanding the implications of computer applications, both from the perspective of the individual and of society as a whole. Students, upon completing this course, should have a broad view of the roles computers can and do play, and an appreciation for the influence computer use can have on social organization.

The material described is content for a half-year course at the secondary level. It is an ambitious course to complete in that time. Many schools, however, will have already integrated some applications into the traditional curriculum. This committee supports and encourages such integration. As applications are integrated into the curriculum, the need for this course may be diminished. Where content of this course duplicates material already integrated, units maybe deleted to allow a more complete treatment of the remaining material. The units presented are not designed to require equal time. Individual school districts and teachers may determine the balance.

Students are not expected to have experience with computers prior to this course. Throughout the course, however, they should have first-hand experience with the computers being used for specific applications. School districts will need to provide sufficient hardware and software so students can spend significant time at the computer. While some computer applications might be displayed through teacher demonstration, the majority are best experienced "hands-on". In this light, the software costs may be relatively high. These software costs will effectively be lower by sharing among other courses as the school integrates computing into the curriculum. In addition to in-class, directed computer use, students should have some free access to computers during an open laboratory time. This could be provided during the school day or after school hours.

Programming is not included in this course. While learning to program allows students to gain control of the computer, they can also gain power in their lives by mastering useful applications. Students should be encouraged to continue working with the machine, using this course as a stepping stone toward gaining mastery of the computer.

School districts should not view this course as the sum of teaching about computers and computing. All students should be given the opportunity to acquire computer skills beyond the level of using application programs. By the same token, all students should be knowledgeable about computer applications and their implications.

Two major themes should be interwoven through all the units: The first is the rapid rate of change in computer technology and use. During this course, a brief look at history and some speculation on future potential should be included with each topic. For example, space exploration is very dependent on computer use and has been able to progress because of computer availability. It will also have its future influenced by the direction of computer science research.

Careers and career potential make up the second theme to consider throughout the course. Career awareness is important to secondary school students; two facets, careers in computing and impact of computers on all careers, should be discussed along with the topics studied during this course. For example, robotics and word processing lend themselves readily to discussion of the impact they will have on careers.

Due to the rapid rate of change in the computer industry, this course will need continual updating. Teachers are urged to use a variety of resources to help them provide up-to-date instruction in this course. Specific suggestions include:

Guest lectures: people who use computers in their professions or hobbies.

Trips to computer installations: different uses require different types and sizes of systems.

Humor: cartoons and jokes convey information as well as lighten discussions.

Novels and stories: speculative fiction displays the meshing of reality and imagination about computers.

Films: many are available, some telling about computers, some demonstrating computer uses such as animation.

Independent student research: both research papers and oral reports to the class are appropriate in this course.

COURSE DESCRIPTION

UNIT 1: Computers and Program Control

OBJECTIVES:
Students will understand that computers receive data, process data, and output data.
They will understand that computers follow directions when processing data.
They will be able to express, in pseudocode or natural language, a simple algorithm.

SUGGESTED APPLICATION:
Many computers have software that gives a "tour" of the machine. The students should use this software now as an introduction to the particular system being used.

UNIT 1 OUTLINE:

I. What a Computer Does
   A. Processes data
   B. Follows directions
   C. Possible sources of errors

II. Social Issues
   A. Who gives the directions
   B. Dealing with errors

UNIT 2: Computer Components and Data Representation

OBJECTIVES:
Students will know the basic parts of a computer system and understand the flow of data among them.
They will be able to list some of the input/output and storage devices currently in use.

SUGGESTED APPLICATION;
There are programs that simulate or make a game of various computer components or methods of data handling. Realizing the depth of understanding the student will have, one of these applications may be appropriate in this unit. If not, many films have been produced to show computer components and may be shown here.

UNIT 2 OUTLINE:

I. Computer Organization
   A. Input devices
   B. Central processing unit
   C. Output devices
   D. Storage for data
   E. Representing data in the computer

II. Social Issues
   A. Rapid rate of change in technology
   B. Future technology trends

UNIT 3: Word Processing

OBJECTIVES:
Students will use the word processor to produce documents for assignments in other classes they may be taking.
They will understand the difference between using a word processor and using a typewriter.
They will be able to format and reformat text for printing.
They will understand the distinction between the composition task and the word processing task.

SUGGESTED APPLICATION:
Students should have "hands-on" experience with a word processor. The processor selected should be easy to learn, with on-line support, yet powerful. The object is to use the word processor, not memorize control codes. It is important to allow the students time to practice using the word processor.

UNIT 3 OUTLINE:

I. Word Processing
   A. Using the computer to enter, manipulate and print text
   B. Benefits to the user

II. Social Issues
   A. Production of needless and excessive materials
   B. Excellence or repetition through revision?

UNIT 4: Robotics and Artificial Intelligence

OBJECTIVES:

Students will recognize the general goals of work in artificial intelligence. They will be exposed to the complexity of the problems researchers in AI are investigating.
Students will realize the impact that intelligent robots could have upon the production of goods.
They will understand the potential for social change due to robots.

SUGGESTED APPLICATION:

Since this will be a brief overview of robotics and artificial intelligence, there may not be applications available to demonstrate the techniques used in AI. If you have a simple robot available at your school, it should be demonstrated in this unit. You may wish to consider using a film in presenting this unit.

UNIT 4 OUTLINE:

I. Introduction to Robotics and Artificial Intelligence
   A. Definitions of Robotics and Artificial Intelligence
   B. Areas of research
   C. Robots and the production of goods
   D. Robots and environments inhospitable to humans

II. Social issues
   A. ELIZA or DOCTOR and Weizenbaum's criticisms
   B. Natural language processing and privacy in human conversations
   C. Threat to human uniqueness
   D. Unemployment due to displacement by robots
   E. Home use
   F. Effect of reduced demands on people
   G. Human/robot relations

UNIT 5: Modeling and Simulation

OBJECTIVES:

Students will understand the advantages of simulating an environment or process.
They will realize that simulations involve the simplification of the environment or process.
They will attempt to model a process for developing a simulation.

SUGGESTED APPLICATION:

Students can use a variety of computer simulations. One should be of a control process (e.g. a nuclear power plant simulation); another of an environment (e.g. a lake, predator populations). Students should look for

simplifications the developer made and discuss the ramifications of any possible mistakes.

UNIT 5 OUTLINE:

I. Modeling and Simulation of Environments or Processes
   A. Developing the model
   B. Implementing the model
   C. Applications of simulations

II. Social Issues
   A. Decisions based upon simulations
   B. Results of an incorrect or oversimplified model
   C. Increased complexity in combining simulations
   D. War games by computer

UNIT 6: Business and Numerical Applications

OBJECTIVES:

Students will understand the changes that computers have brought or could bring to the business world.
They will learn to use an electronic spreadsheet for simple manipulation of data and what/if calculations.
They will be aware of the uses of computers in analyzing scientific data.

SUGGESTED APPLICATION:

The students should gain experience in using a simple spreadsheet, applying it to a problem meaningful to them (e.g. how grades this term will effect their grade average or using it to analyze data from a science class they may be taking)

UNIT 6 OUTLINE:

I. Numerical and Business Applications
   A. Information processing in business
   B. Financial planning and accounting
   C. Scientific analysis applications

II. Social Issues
   A. Security of data
   B. Accuracy of data

UNIT 7: Computers and Communication

OBJECTIVES:

Students will realize that the combination of computer and telecommunications represent a new form of communication.
They will be able to compare and contrast computer communications with other forms of communication.
They will have a basic understanding of electronic funds transfer and electronic mail.
They will recognize some possible changes in society that this technology may bring.

SUGGESTED APPLICATION:
With a modem and appropriate software, demonstrate a local bulletin board. If possible, allow the students to leave and exchange messages on the board.

UNIT 7 OUTLINE:

I. Communicating Information with Computers
   A. Applications of remote communication
   B. Hardware and technology considerations

II. Social Issues
   A. Data verification
   B. Cashless society
   C. Crimes and fraud
   D. "Electronic cottage"
   E. "Electronic hermit"
   F. Access to systems by unauthorized persons
   G. Contrast to other communication methods

UNIT 8: Public Information Systems

OBJECTIVES:
Students will be aware of the variety of information systems available to the public and what information is available.
They will understand the basic boolean logic forms and the use of them in searching a database for information.
They will devise a search strategy for research in a subject area.
They will appreciate the changing nature of accessing information.

SUGGESTED APPLICATION:
Demonstrate accessing an information service. Have the students plan the search strategy. If your library has search facilities, assign each student to do a search of interest to them.

UNIT 8 OUTLINE:

I. Public Information Through the Computer
   A. Information resources
   B. How to access information
II. Social Issues
   A. Information cost
   B. Equity of access
   C. Potential changes in human use of information
   D. Potential changes in libraries
   E. Electronic publishing and copyrights

UNIT 9: Private Information Data Banks

OBJECTIVES:
Students will understand that business and government agencies have a need to keep information on file.
They will recognize that the amount of information stored is rapidly growing.
They will understand the need for restricted access to information files.
They will be able to compare and contrast the public's/government's right to know about an individual and the individual's right to privacy.
They will be able to use a database program to enter information, sort the information and format lists of information.

SUGGESTED APPLICATION:
Have the students use a database (filer) program to construct a data bank of use to them (e.g. mailing lists, bibliographic information). Choose a filer that is easy to enter data, sort, and format lists.

UNIT 9 OUTLINE:

I. Data banks
   - A. Organization of information and access
   - B. Varieties of data banks

II. Social Issues
   - A. The "public" need to know
   - B. Privacy, the individual's right
   - C. Unauthorized access
   - D. Individual right to check and change erroneous information
   - E. Connecting data banks

UNIT 10: Graphics

OBJECTIVES:
Students will understand the importance of graphic display of information in a wide variety of applications.
They will gain experience using a business graphics package and a graphics utility package.

SUGGESTED APPLICATION:
Using the data the students manipulated in the spreadsheet unit, have them use an application package to display it in a graph. Require at least two different types of graphs (e.g. pie, bar, line). As a second exercise, use a graphics package designed for home use. If possible, use one that includes a graphics tablet or light pen

UNIT 10 OUTLINE:

I. Graphics Applications
   - A. Applications of graphics
   - B. Hardware and technology considerations

II. Social Issues
   - A. Changes in entertainment
   - B. New ways to visualize data

AUTHORS OF THE REPORT

Fred Achberger is Computer Specialist for the Computer Information Center at the Bremerton, Washington, Educational Services District and for the Bremerton School District. His 12 years teaching experience include the elementary as well as secondary level. He has a B.S. from Western Washington College and is working on an M.S. through St. Martins College.

Robert M. Aiken, currently at Temple University, has been active in computer science education for many years, at the national and international level as well as through the university. He is author or co-author of numerous books, including the Computer Power series. His Ph.D. is from Northwestern University.

John C. Arch, now at Shippensburg State University, taught for 10 years in public secondary schools. He has written the NEA Federal Tax Guide for teachers, and the Computer Literacy Curriculum for the Houston Independent School District when he was secondary school computer science specialist there. He has a J.D. degree from Vanderbilt as well as a Ph.D. from the University of Oregon.

Michael R. Haney, of Towson State University, Maryland, has extensive experience with pre-college teacher training as well as 12 years secondary school teaching experience. He is contributing author of Computer Applications for Science Teachers and is now working on a methods book for high school computer science teachers. His Ph.D. is from the University of Oregon.

John D. Lawson, Jr., a teacher and doctoral student at the University of Oregon, has also taught at the secondary school level. He helped develop and implement a vocational training program for Native Americans in Canada. He has an M.S. in Teaching Geography and an M.S. in Computer Science Education from the University of Oregon.

Cheryl Lemke is currently Specialist in the Computer and Technology unit, office of the Superintendent of Public Instruction, Washington State, after having taught in public secondary schools for 9 years. She is active in state, regional and national organizations for mathematics and computer science teachers, and has a B.S. from Western Michigan University and Fifth Year Credential from Central Washington University.

Jean B. Rogers, University of Texas, Austin, has been working in pre-college computing and computer science for 9 years. Author of a variety of publications on this topic, she is currently Chair of the ACM Elementary and Secondary Schools Subcommittee. Her Ph.D. is from the University of Oregon.

Thomas A. Swanson, now at Jefferson High School in Portland, OR, has taught mathematicsand computer science at the secondary level for over 11 yrs. He has an M.A. from the University of Oregon and is currently a doctoral candidate in the Computers in Education program at that university.

Samuel F. Tumolo, is Chair of the Department of Mathematics, Cincinnati Country Day School. He has taught at the secondary school level for 15 yrs., including 8 teaching computer science courses. He has an M.S. from Xavier University in Mathematics and an M.S. from University of Oregon in Computer Science Education.

REFERENCES

ACM Curriculum Committee on Computer Science. Revised recommendations for CS 1. Communications ACM in press.

ACM Elementary and Secondary Schools Subcommittee. Topics: computer education for elementary and secondary schools. Association for Computing Machinery: New York, 1981.

ACM Education Board. ACM curricula: recommendations for computer science (V.1) for information systems (V.2) for related computer science programs (V.3). Association for Computing Machinery: New York, 1983.

College Board's Advanced Placement Computer Science Development Committee. Teacher's guide to advanced placement courses in computer science. College Entrance Examination Board: Princeton, NJ, 1983.

IEEE Model Program Committee, Educational Activities Board. Model program in computer science and engineering. Institute of Electrical and Electronic Engineers: Silver Spring, MD, 1983.

Victorian Institute of Secondary Education. Computer Science High School Certificate Course Description. Victorian Institute of Secondary Education: Melbourne, 1983.

THE REPORT OF THE ICTM SC3

(SECONDARY COMPUTER CURRICULUM COMMITTEE)

The contents of this report are not copyrighted and may be quoted or reprinted without formality other than the the customary acknowledgment of the source. A letter to the ICTM Treasurer (address at end of report) should be sent indicating that a quote has been used or copies have been made.

FOREWORD - By: John W. McConnell, President ICTM 1984-1985

In 1983, the Illinois Council of Teachers of Mathematics amended its By-Laws to extend the responsibility of our organization to the area of computer education. The major reasons for this expansion of the ICTM mission included the almost universal tendency in Illinois schools for mathematics teachers to be leaders in installing computer education programs and the natural relationship between computer science and mathematics. Even before the change in its By-Laws, ICTM had taken a leadership role in encouraging the use of computers throughout the school curriculum. ICTM published its first major document on computers, "Report of the Task Force on Microcomputers in the Schools," in 1981. That report itemized a variety of uses of computers as problem solving tools, as objects of study, and as teaching aids. The report anticipated that each of the areas would increase in importance and deserve further study as Illinois schools installed more microcomputers.

The establishment of the Advanced Placement Computer Science Course by the College Board in 1983-84 gave the subject of computer programming a content focus which it did not have previously. Simultaneously, school districts across Illinois were experimenting with computer courses at a variety of grade levels. Publishers of software have produced software packages which bring important applications such as word-processing, spreadsheets, and data-base management to the school environment. Again, mathematics teachers have played a major role incorporating AP computer science, programming classes, and microcomputer applications into their schools. All of these factors increased the need to study computer education as a new and important curriculum in Illinois schools.

The Secondary Computer Curriculum Committee was given the charge to assimilate the many different trends and issues, and produce a document which would help Illinois schools identify those experiences and outcomes which would define a computer curriculum. Committee members spent many hours reading computer articles and documents, writing position papers for committee study, arguing and discussing the major issues, and writing the document in this pamphlet. Under the leadership of co-Chairs Lowell Carmony and Don Enns, the committee has wisely limited the scope of its recommendations to the general curriculum of the school. This document is very important for education because of its delineation of the important cognitive skills computer experience can enhance. Students who go through the courses described by the committee should be better prepared for further work in computer science; better prepared for vocational computing courses; more adept at using the computer to solve problems; abler at communicating a solution to a problem; and more experienced in important uses of computers.

This document will be valuable for teachers who need support in their efforts to establish a computer curriculum in their schools. It will help junior high schools and high schools define a sequence of computing skills and improve articulation among grade levels. The committee has encouraged applications and cooperation across subject area boundaries. The course outlines will help college and universities and the Illinois Computer Consortia define inservice which will qualify instructors to teach these courses.

The committee is to be highly praised for skillfully defining this new curricular area. I commend the report to you, and encourage you to disseminate its recommendations to your colleagues and administration.

THE REPORT OF THE ICTM SC_3
(SECONDARY COMPUTER CURRICULUM COMMITTEE)

INTRODUCTION

It is evident that there is a growing curriculum in computer science at the secondary level. Some schools have been teaching isolated courses for more than a decade while other schools are just now beginning to implement a program in computer science. To aid schools in Illinois in the development of a full program in computer science, the Illinois Council of Teachers of Mathematics charged its Secondary Computer Curriculum Committee with the task of recommending a computer curriculum. This report contains the outlines of the two recommended year long courses in computer science designed by that committee. The committee focused on the needs of the high schools, but recognized that many Illinois junior high schools are ready to install portions of the first course.

The first course, entitled "**Introduction to Computing**", is designed as an introductory course for all interested students. This course presents the major applications of computers such as word processing, data base management, and spreadsheets, discusses the historical and societal impact of computers, and introduces the student to programming. The second course, "**Computer Science I and II**", is intended as a serious introduction to structured programming for any college bound student interested in computers. This second course should be viewed as a course in problem solving on a computer. For those schools with sufficient interest, a third course built around the new Advanced Placement course in computer science can be offered for those students planning to major in computer related fields at the university level. Because the AP course is well defined, an outline has not been provided here. For those schools with the resources and the students, an assembly language course can also be added to the curriculum. An outline for such a course has not been included here because at present few schools in Illinois have computers with a sufficiently powerful instruction set and few schools have the staff trained for such a course.

Computer Science is more than language acquisition. Hence, the committee strongly urges schools not to offer sequences of courses in a general curriculum with a semester in language A, then a semester in language B, and then another in language C, etc. The committee encourages schools to teach one language within each year sequence so that students can gain depth in understanding of computing principles. Teachers should not wait for the "perfect" computer language for these courses. Educators must be aware that languages are themselves evolving. The committee does, however, expect that Logo, BASIC, FORTRAN, PL1, and Pascal will be the dominant languages of the near future. Schools that choose to continue with FORTRAN or BASIC should strive to obtain the most structured versions of these languages available. When an older version is used, care must be taken to preserve a structured approach to programming. For example, students should be required to declare their variables through remarks if declarations are not required by the language. Likewise the use of the **GOTO**, if necessary in the language, should be restricted to a few particular situations such as the emulation of an If available and these may be adequate even for the second course of the enclosed outline. Rather than dodge the question of language, it is the intent of the committee to point out that computer science is so much more than language. Language issues will continue to be hotly debated, but in the meantime schools

should strive to implement the most modern program in computer science that their resources and staffing allow.

The committee does not recommend that either of its courses be required of all students. Computer science is viewed by the committee as a popular elective at the secondary level. Many students will choose to enroll in at least one semester of a course in computers, but we feel that little is to be gained by requiring computer science of all students at the secondary level.

School administrators should be aware that the courses described here do not constitute the entire instructional computer curriculum. Computer study includes computer aided instruction, vocational training, and computer science theory. This committee considered only the liberal arts curriculum, and, hence, computer science theory is the area addressed by this document. Computer aided instruction and vocational training create added demands for computer equipment, supplies, staff, software, instructional time and training. The availability of vocational funds have led some school districts to expand vocational computer training at the expense of the general computer curriculum. We encourage the development of a balanced curriculum containing both vocational as well as computer science theory courses. Many schools offer COBOL as part of their vocational data processing programs. The first year sequence would be excellent preparation for vocational courses in computing.

Schools planning to implement new computer curricula should realize that teacher preparation is as important as the hardware and software necessary for the courses. Since computer science is a new subject, many teachers will need inservice assistance in order to implement the courses listed here. Administrators must plan in advance which personnel will take the primary responsibilities in the program and how these persons will be trained. Some regional computer consortia are able to provide resources for staff inservice. Illinois universities and colleges are establishing courses appropriate for computer teachers.

RECOMMENDATIONS FOR A GENERAL CURRICULUM INTRODUCTORY COURSE

INTRODUCTION

The topics for the general curriculum include introductory programming concepts, applications of computers, history, ethics, and social implications. Most schools will find it convenient to develop an elective course around these topics. It is recognized that schools in Illinois will use a variety of methods to present instruction in these introductory areas. The intended audience for the course described here is students in the average or general curriculum at the high school. The topics can be integrated into a sequence leading to higher level courses. Some school districts may wish to incorporate some of these topics into the junior high/middle school curriculum.

It is important that all computer technology courses address the moral and ethical aspects related to the use of computers. Courses which enroll adolescents and younger students should provide a clear definition of personal responsibility and respect for the rights of others. The study of ethical issues should be integrated throughout the curriculum.

No topic listed in the outline for the general computer course curriculum requires a specific programming language or particular brand of computer equipment. Most computer equipment available in schools and homes comes with a version of the BASIC (Beginners' All-purpose Symbolic Instructional Code) language. Other languages which are widely available and appropriate for beginners include simple LOGO and versions of interpreted Pascal. Even though BASIC has been criticized by computer scientists as being unstructured and pedagogically unsound, BASIC will not disappear from the computing scene. Its ease of use, enhanced by new versions which will incorporate the powerful structures advocated by computer scientists, will insure its survival for many years. The issue in introductory courses is not whether the language should be "structured," but rather how we can help students structure their solutions to a problem.

Structured programming skills facilitate the development of problem solving skills. A programming course should give adolescents strategies for problem solving which will transfer to other situations. The computer language by itself does not guarantee good problem solving skills, just as a knowledge of English does not guarantee good writing skills. No language, structured or unstructured, is sufficient to develop logical thought processes. The teacher is the most important facilitator in the intellectual growth of children. The teacher has the difficult responsibility of insuring that student programs communicate the solution of a problem to other human beings. Some of the areas in which the computer teacher plays a significant role in developing higher order intellectual skills follow:

ANALYSIS. The student should analyze the problem before coding. An important step in the solution of a problem is breaking the problem into smaller components. The teacher must provide examples of problem analysis which illustrate the technique to students. The teacher must tame the student's tendency to compose programs at the keyboard. The teacher should present an organizational method which helps the student think through the problem before doing any translation to a computer language. That method might be a flow-chart of

major steps, a paragraph description, or some other written communication of the student's analysis.

TRANSLATION. The student should translate the component parts of the analysis into computer code. It is easy to view the coding as the primary goal of an introductory computing course. The desired end result, however, is growth in thinking skills, not just the learning of a language like Pascal or BASIC.

REFINEMENT. The student must synthesize the component parts of the solution into a computer program which has logical integrity. The teacher must help the student go beyond just "getting it to work." It is the teacher's responsibility to help the student make his computing product agree with his analysis.

COMMUNICATION SKILLS. The result of the problem solving activity should be a communication to others. The inherent fascination with computers makes it easy for students to communicate with the machines. The teacher must require that the student submit program listings, output, and documentation in a form which clearly communicates the problem solution.

The usefulness of the computer should be demonstrated to the students. Applications broaden a student's concept of programming. Spreadsheet and data base programs, for example, require an understanding of data structures and operations. Experience with a variety of applications helps the student to generalize the concepts of computer science.

It is through applications that schools can expand the use of computers throughout the curriculum. Some schools may choose to assign to various departments the responsibility of teaching some aspects of computing. For example, the positive and negative effects of computers on society could be a topic in a social studies course. A word processing program could be a major tool in English classes. Programs which deal with graphics might be integrated into art or drafting courses.

The major components of this course are applications, simulations, programming, and history and ethics. Applications provide opportunities for students to experience problem solving in a computer context. The skills developed through the use of applications packages are analogous to the skills used in programming. Application packages allow students to work with more realistic problems and data than has heretofore been possible with only textbook, pencil and paper. Simulations can be used to bridge the gap between applications and programming in a computer language. Simulations allow for the mental processes of problem solving without the detailed mechanics of programming or application formulas. Programming gives the student more powerful techniques to solve problems. The language, algorithms, and data structures developed in the learning of a language allow the student to use the computer as an "intelligence amplifier." History and ethical considerations should be included in the course to place the power of the computer in perspective. As students work with equipment, the teacher must maintain procedures which promote respect for the rights of others and which

help students accept personal responsibility for using the computer in proper ways. In many cases, teachers will find it appropriate to interleave these technical topics throughout the course so that students will reach deeper levels of understanding about each facet of computing.

COURSE OUTLINE
INTRODUCTION TO COMPUTING

Prerequisite: None.
This course is intended as an introduction to the concepts of computing. Some schools currently require that students have algebra before they qualify for a computing course. That is not the case here. This course will present and develop the concept of variables and their importance in computing, order of operations, logical sequencing, and coordinates in graphing.

Length of course: This course is planned to be two semesters in length. However, with suitably prepared students, the information could be covered in one semester.

I. **APPLICATIONS** (Using programmed software and commercial packages)
   - A. Word Processing
     1. Entering text
     2. Editing text
     3. Printing
     4. Storing text
   - B. Data Base
     1. Format (records, fields, data types)
     2. Manipulation
        - a. Entering records
        - b. Editing records
        - c. Selecting and Sorting records
        - d. Storing and retrieving information
     3. Reports
   - C. Spreadsheets
     1. Format (Columns, rows, formulas, data types)
        - a. Creating a model
        - b. Layout and structure
     2. Manipulation
        - a. Entering data and formulas
        - b. Editing data and models
     3. Graphical summaries

II. **SIMULATIONS** (The bridge between applications and programming.)
   - A. Using existing packages and programs
   - B. Using spreadsheets for modeling
   - C. Creating simulations in a programming language

III. **PROGRAMMING**
   - A. Documentation
   - B. Input
     1. Interactive
     2. Non-interactive

C. Processes
   1. Computational
   2. Decision-making (Logical)
   3. Repetition (Loops)
   4. Subroutines
D. Output
   1. Text
   2. Graphics

IV. **HISTORY AND ETHICS**
A. Development of hardware and software
B. People in the history of computing
C. Uses and misuses of the computer
D. Effects of computer technology on society
E. Personal responsibility

RECOMMENDATIONS FOR COURSES IN COMPUTER SCIENCE

INTRODUCTION

Computer Science is a two-semester course in problem solving, not language syntax. One must use a computer language to communicate one's algorithm to the computer, but the main course objective has to be the acquisition of general problem solving skills. Through examples, students must be shown general problem solving strategies and be led to develop their own skills in these areas. Students should learn to view problem solving as a four-step process:

1. **Problem Analysis.** The first step in problem solving is to understand the problem. Until the problem is understood, the student has no reason to touch the computer. A student's understanding of a problem is often enhanced by working through several "test" cases by hand.

2. **Algorithm Design.** Using a top-down design, divide and conquer the problem through a sequence of modules. This step can use flowcharts or pseudo-code, but does not include programming. Students should test their design and refine it by using the test cases from step 1. Repeat the analysis and design steps until convinced that a reasonable algorithm has been found.

3. **Implementation.** Code the design in the language of your choice. Note that this is the first programming step and should be fairly routine if steps 1 and 2 have been done properly.

4. **Verification and Modification.** Does the program solve the given problem? If not, repeat steps 1 to 3.

Another concern that should permeate computer science courses is the need for clear, readable programs. Students should understand that programs are written to be read by humans as well as by machines. Therefore the same attention to clarity that one would expect in a composition class should be given in a programming class. Programs should contain internal documentation with remarks or comments. Large projects should also have external documentation in the form of a "user's manual." Meaningful variable names and indentation in programs should be common practice.

In this course, attention should also be given to the creation of well-articulated, independent modules that together solve the given problem. The advantages of local variables over global variables should be discussed. The teacher should discourage the use of global variables and encourage the use of local variables and passing of parameters. As in the introductory course, the language alone will not produce good programming habits. The teacher must provide the examples of problem analysis, algorithm designed based on modules, implementation, and verification and modification.

COURSE OUTLINE
COMPUTER SCIENCE I

Prerequisite: Algebra I
Some skills in symbol manipulation and the concept of variable are needed for this course. The pace of delivery requires a maturity level commensurate with that of a student who has an Algebra I background.

Length of course: One semester.

I. **Introduction**
   A. History
      1. Computing
      2. Languages - high/low level
      3. Computer evolution - generations
   B. Terminology
      1. Hardware/Software - CPU, program, etc.
      2. Storage Media
      3. Compiler/Interpreter

II. **Operating the Computer**
   A. Accessing Programs and Data
      1. Internal/External Memory
         a. volatile such as RAM
         b. non-volatile such as disk based storage or ROM
      2. Handling Considerations
         a. proper care of diskettes
         b. responsible use of hardware
   B. Shared Storage Resources
      1. Security of data and programs
      2. Privacy of information
      3. Responsible use, ethics and legal implications
   C. Specific Operating Systems - installation dependent

III. **PROGRAMMING**
   A. Simple Data Types (as available)
      1. Integer
      2. Real - internal representation and accuracy
      3. Character
      4. Boolean
      5. String
   B. Variable and Constant Declarations
      (Necessary for structured programming and should be supplied through comments even if not required by the language)
   C. Assignment
      1. Arithmetical and logical operators
      2. Precedence of operators
   D. Input/Output
      1. Formatting
      2. Redirecting of input and output files
   E. Selection (Methods of modelling partitioning processes)
      1. Logical conditions
      2. If - Then - Else
      3. Case (as available)

F. Repetition (Methods of modelling iterative processes)
    1. While (pre-test)
    2. Repeat (post-test)
    3. For (automatic indexing)
G. Subprograms to Support Modular Structure
    1. Procedures
        a. built-in
        b. user-defined
        c. parameters
            i. passed by value
            ii. passed by variable - address
    2. Functions
        a. built-in
        b. user-defined
    3. Scope - local/global identifiers
H. User Defined Types (as available)
    1. Subranges
    2. Enumerated types
I. Arrays
    1. One dimensional lists
    2. Multi-dimensional tables
J. Numerical Analysis
    1. Accuracy in computation
    2. Root finding
    3. Area under a curve
K. Graphics (as available) to enhance problem solving

COURSE OUTLINE
COMPUTER SCIENCE II

Prerequisite: Computer Science I
The purpose of this semester course is to reinforce and extend the students' knowledge of programming and its applications. The general considerations of Computer Science I, such as structured programming, modularity and clarity, are even more important in Computer Science II because of the increased complexity of the programs and concepts. Computer Science II extends the language elements and applications of Computer Science I.

Length of course: One semester.

I. **INTRODUCTION**
- A. Motivating the Need for Data Structures
  1. Ease of data manipulation depends upon structure of the data
  2. Simplicity of the algorithm is determined by the appropriate choice of a data structure
  3. Common applications require highly organized data
- B. Review of the Benefits of the Structured Approach

II. **PROGRAMMING**
- A. Searching Arrays
  1. Linear search
  2. Binary search of a sorted array
- B. Sorting Arrays
  1. Selection Sort
  2. Bubble Sort
  3. Insertion Sort
  4. Shell Sort (optional)
- C. Sets
  1. As a filter (could be done earlier)
  2. Operations and relations
- D. Records
  1. Fixed
  2. Variant (optional)
- E. Files
  1. Internal - array of records
  2. External - disk based
     a. text
     b. sequential access
     c. random access (optional)
     d. merge sorting (optional)
- F. Linked Lists
  1. Pointer variables
  2. Insertion and deletion
  3. Insertion sorting
  4. Traversing list
- G. Stacks and Queues
  1. Array implementations
  2. Linked list implementations
  3. Operations on stacks and queues

H. Recursion
   1. Applications to problem solving
   2. Understanding recursion via stacked environments
   3. Illustration through graphics
I. Binary Trees (optional)
   1. Construction of trees
   2. Binary search trees
   3. Traversing trees using recursion
J. Applications, Simulations and Projects
   1. Case studies in a variety of fields
   2. System design and implementation
      a. by an individual
      b. by a team

LIST OF RESOURCES

1. PRINT MEDIA

THE COMPUTING TEACHER (Journal of ICCE)
University of Oregon, 1787 Agate Street, Eugene, OR 97403 - 1923
503-686-4414

ELECTRONIC LEARNING
PO Box 644, Lyndhurst, NJ 07071-9985

CLASSROOM COMPUTER LEARNING
Peter Li, Inc., 2451 East River Road, Dayton, OH 45439

INFOWORLD
1060 Marsh Road, Suite C-200, Menlo Park, CA 94025

CREATIVE COMPUTING
PO Box 5214, Boulder, CO 80321

COURSEWARE IN THE CLASSROOM
Ann Lathrop and Bobby Goodson
Addison-Wesley, Reading, MA 1983

DON'T: OR HOW TO CARE FOR YOUR COMPUTER
Rodney Zaks
Sybex, Berkeley, CA 1981

COMPUTER LITERACY: A HANDS-ON APPROACH
Herbert Peckham
McGraw-Hill, New York 1983

COMPUTER POWER: A FIRST COURSE IN USING THE COMPUTER
Michael Moshell, Project Director
Gregg/McGraw-Hill, New York 1982

ACM CURRICULA RECOMMENDATIONS 1968-1983:
Volume I. Computer Sciences
Volume II. Information Systems
Volume III. Vocational-Technical Schools, Community and Junior Colleges, and Health Computing
Association of Computing Machinery

SIGCSE BULLETIN
Special Interest Group on Computer Science Education
Association of Computing Machinery

JOURNAL OF COMPUTERS IN MATHEMATICS AND SCIENCE TEACHING
Association for Computers in Mathematics and Science Teaching
P.O. Box 4455, Austin, TX 78765

EDUCATIONAL TECHNOLOGY
Educational Technology Publications
720 Palisade Avenue, Englewood Cliffs, NJ 07632

COMPUTERS IN MATHEMATICS EDUCATION
1984 Yearbook of the National Council of Teachers of Mathematics

ABACUS
Springer-Verlag, 175 Fifth Avenue, New York, NY 10010

2. ELECTRONIC MEDIA

The Computer Programme
BBC. Available through Time-Life Films

Engineering Renewal and Growth Program
(Video courses in Ada, Fortran, Pascal, BASIC)
Colorado State University
Fort Collins, Colorado 80523

3. ORGANIZATIONS

Association for Computing Machinery
P.O. Box 9182, Church Street Station
New York, New York 10249

Minnesota Educational Computing Corporation (MECC)
3490 Lexington Avenue North, Saint Paul, MN 55112
612/481-3500

International Council for Computers in Education (ICCE)
see THE COMPUTING TEACHER (above)

Total Information Educational Systems (TIES)
Minnesota School Districts Data Processing Joint Board
1925 West County Road B2, Saint Paul, MN 55113

National Council of Teachers of Mathematics
1906 Association Drive, Reston, VA 22091

The College Board
Advanced Placement Program
AP Services, CN 6671, Princeton, NJ 08541-6671
609/734-3800

4. ILLINOIS COMPUTER CONSORTIA

STATE COORDINATOR
Dr. Ray Schaljo 217/782-2826
Illinois State Board of Education
100 North First Street
Springfield, IL 62777

1. **Northwest Illinois Computer Consortium** (Jo Daviess, Carroll, Stephenson, Winnebago, Boone, McHenry, Ogle, DeKalb Counties). Mr. Frank Witt, Director. Courthouse, Room 712, Rockford, IL 61101 815/987-3060.

2. **Lake County Computer Technology Consortium.** Dr. Allen David, Director. 19525 W. Washington, Grayslake, IL 60030. 312/223-6681.

3. **MICRO-IDEAS** (North Cook). Mr. Richard Nelson, Director. 2701 Central Rd., Glenview, IL 60025 312/998-5065.

4. **CADRE** (DuPage, Kane). Mr. Michael Gordon, Director. 570 Pleasant Avenue, Glen Ellyn, IL 60137 312/790-0330.

5. **Micro Partners** (West Cook). Mr. James Weidman, Director. Elmwood Park High School, 8201 West Fullerton Avenue, Elmwood Park, IL 60635. 312/452-9750.

6. **Computer Technology Services Consortium** (Chicago). Ms. Terri von Schaumburg, Director, Chicago Board of Education, 1819 West Pershing Road, Chicago, IL 60690. 312/890-8665.

7. **South Cook County Computer Consortium.** Dr. Brigitte Erbe, Director. Blackhawk--Room 108, 375 Oswego, Park Forest, IL 60466. 312/481-2530.

8. **West Central Illinois Computer Consortium** (Rock Island, Whiteside, Lee, Henry). Mr. Mike Williams, Director. Merrill Elementary School, 604 4th Avenue, Rock Falls, IL 61071. 815/625-1495.

9. **Illinois Valley Computer Consortium** (Bureau, LaSalle, Putnam, Marshall, Woodford). Mr. Joe Benning, Director. 119 West Madison-Room 102, Downtown Courthouse, Ottawa, IL 61350. 815/433-9495.

10. **Tri-County Computer Consortium** (Kendall, Will, Grundy). Mr. John Flynn, Director. Lewis Joliet Renaissance Center, 214 North Ottawa, Joliet, IL 60431. 815/727-6544, Ext. 301.

11. **Western Illinois Computer Consortium** (Mercer, Warren, Henderson, Hancock, McDonough, Adams, Schuyler, Brown, Cass, Pike, Morgan, Scott, Greene, Calhoun). Mr. Bob Rigney, Director. 1444 Maine, Quincy, IL 62301. 217/223-8700, Ext. 53.

12. **Mid-Illini Computer Consortium** (Knox, Peoria, Fulton, Tazewell). Ms. Louise Bassett, Director. Fulton County Courthouse, P.O. Box 307, Lewistown, IL 61542. 302/547-3041, Ext. 270.

13. **Abe Lincoln Computer Consortium** (Menard, Sangamon, Montgomery). Ms. Elsie M. Stoeckel, Director. 1300 North 11th Street, Room 178, Springfield, IL 62702. 217/525-2002.

14. **Illinois Heartland School Computer Consortium** (McLean, Mason, Logan, DeWitt, Macon, Christian). Mr. Paul Burton, Director. 2240 East Geddes, Decatur, IL 62526. 217/424-3434.

15. **East Central Illinois Micro-Computer Consortium** (Livingston, Kankakee, Iroquois, Ford, Champaign, Vermillion, Platt, Douglas). Ms. Debbie Harshbarger, Director. 200 South Frederick Street, Box 959, Rantoul, IL 61866. 217/893-8481.

16. **TENCO/SURSKA Microcomputer Assistance Consortium** (Macoupin, Jersey, Madison, Bond, Clinton, St. Clair, Washington, Monroe, Randolph). Mr. Jed Deets, Director. Educational Developmental Center, 1505 Caseyville Avenue, Belleville, IL 62221. 618/277-4530.

17. **Mid-Central Illinois Computer Alliance** (Edgar, Moultre, Coles, Shelby, Cumberland, Clark, Fayette, Effingham). Mr. Mark Myers, Director. Lake Land Community College, South Route 45, Mattoon, IL 61938. 217/235-3131.

18. **Computer Technology Services Consortium** (Jasper, Crawford, Marion, Clay, Richland, Lawrence, Jefferson, Wayne, Edwards, Wabash, Hamilton, White). Ms. Kris Wilson, Director. Box 277, Fairfield, IL 62837. 618/847-3151.

19. **Computer Technology Services Consortium** (Perry, Franklin, Jackson, Williamson, Saline, Gallantin, Union, Johnson, Pope, Hardin, Alexander, Pulaski, Massac). Mr. John Gardner, Director. c/o Courthouse, Harrisburg, IL 62946. 618/253-5581.

20. **Department of Corrections Computer Consortium.** Mr. Bill Choat, Director. 1301 Concordia Court, Springfield, IL 62702. 217/522-2666.

MEMBERS OF THE COMMITTEE

Lowell Carmony (Co-Chair)	Lake Forest College Lake Forest
Don Enns (Co-Chair)	Highland Park High School Highland Park
Jerry Cummins	Proviso West High School Hillside
Greg Gruenkemeyer	Salem High School Salem
Bernard Kelly	Forest View High School Arlington Heights
John McConnell	Glenbrook South High School Glenview
Bill Messersmith	Rich South High School Richton Park
Rick Nelson	Micro-Ideas Glenview
Billie Porter	Glenbrook North High School Northbrook
Ray Schaljo	State Board of Education Springfield

ICTM GOVERNING BOARD

1984-1985

John W. McConnell	President
Paul Trafton	Past President
Paula Filliman	President Elect
Darrell Ropp	Board Chairman
Peggy Olson	Secretary
Hal Anderson	Treasurer
Alison Claus	Director, Grades K-3
Karla McDonald	Director, Grades 5-8
Dorothy Ellis	Director, Grades 6-9
Larry Montgomery	Director, Grades 9-12
Chuck Hamberg	Director, Grades 9-12
Carole Bauer	Director, Community College
Larry Sowder	Director, Senior College
Sharon Allen	Director, Special Education
Jerry Becker	Director-at-Large
Sandra Spalt	Director-at-Large

Membership information may be acquired from the Treasurer, Hal Anderson, Mathematics Department, Eastern Illinois University, Charleston, IL 61920

2/14/85

ASSESSING LIBRARY MICROCOMPUTER NEEDS

By: Eric S. Anderson, Micro Computer Libraries, Freeport, IL

One of the things that really sticks in my craw is the hype surrounding libraries implementing the microcomputer into their management schemes. In some ways I suppose I am part of the problem as a long time advocate of micro based library automation: but in other ways the commercial world has come too far and too fast and left too many people behind and sold too many people systems they felt compelled to buy to keep up with the Kremplers. If videotapes could have produced overdue notices they probably would have been guilty as well.

Of all of the tools I have ever had to manage my library, the micro far outdistances all paper based technology combined. I also have no problem making the psychological leap necessary to use a word processor, a spreadsheet and a data base, and, indeed, I've learned to use perhaps a dozen word processors, half a dozen spreadsheets and at least two dozen data bases. I want to see if the latest thing will do something better or give me a different sort or macros or a zillion neat features. This level of investigation is unnecessary for most librarians, and uncalled for. Find something that works in your library and use it.

The trouble arises when folks go to meetings and hear about the joys of micro based circulation for example. Notice I said joys--the pains, the troubles, the crashed data disks and all the rest don't seem to exist, especially in vendors' demonstrations. It's easy to like something that's easy--no one seems to remember that your shelf list doesn't jump into the micro. Somebody has to key it in. Somebody, at least in the words of my sainted father, usually means "me".

Implementing the micro into your scheme of library management needs a plan and it also needs time to mature. The plan comes from those most pressing needs. The time is used for you to develop your own micro skills. If you tackle little projects, gradually moving to larger and larger ones, no one will need explain field lengths and sorts and all of the other components of a micro based circ system. You can ask intelligent questions and make rationale decisions from your own experience, and that's one heck of a lot more meaningful than what you hear at a meeting or see in a demonstration.

We also tend to look to the future with rose colored glasses. We choose as our first priority replacing a card file of 30,000 local newspaper citations and then get so frustrated when we run out of space on a twenty meg drive that we begged and pleaded and whined to get. Bonkers. Absolutely screaming yellow bonkers! Yes, the future is cheaper hard storage and marvelous Mac Networks, but until it gets here we best use our time developing our skills so that when it arrives we know what we can do with it and what we will need to get the job done.

Little pieces, little tasks done on hardware borrowed from the Math department if we have to. Teaching yourself what a word processor is and the qualities of a good screen editor. Watching as the spreadsheet crunches numbers like Darwin classifying moths and wondering why the circulation program you just looked at will only report how many books were checked out last month rather than ranges and percentages and all the rest of the math stuff the micro loves

to do. Use a data base to inventory your AV hardware and track your film bookings or whatever else is small and manageable. Buy a cheap and dirty overdue program and see if it handles your problem and what it's strong suit is but more importantly learn what a 28 character title field means. Teach yourself what the software won't do so as you ready yourself to make a step up you make that step wisely.

Software, short and simple, is your salvation. There are very few school districts without micros these days, and more than likely the librarian is charged with their care. Several of your staff cohorts have more than likely dabbled with a word processor. If you haven't had the privilege, ask them to teach you to use it. Hackers for the most part enjoy sharing the faith, and those that don't aren't hackers--they are in it for the money.

Pick something nice and easy. Make a bibliography of all your materials on the Faulklands (or anything else that suits your fancy). Notice how easy it is to delete a title that sprouts legs and walks away from your collect. Smile as the book order comes in and you get to add two titles and then let the computer retype it for you. If you can walk and chew gum, you are beginning to get the idea.

Before long you will wish you could produce not only a call number list, but a title list. You cry when you find that your word processor doesn't do indexes. You want to sort and the word processor won't do it. Too bad, so sad. What likes to sort? A data base! If you are lucky enough you can bring the word processing file over to the data base without rekeying it. This gets a little fancy, but don't hesitate to ask the hacking crowd. You might even care to ask a student to lend a hand.

The data base sorts just fine, and with the push of a few keys you can get all manner of lists out. Unfortunately, the data base requires you specify how long the field will print and you are continually running long and short of what you thought was right. Another light should be going on about this time. What works well as an inhouse list of sound filmstrip projectors looks absolutely tacky next to the finely formatted word processing list.

You are now ready to look at a piece of library software designed to do bibliographies. Does it contain the fields you use? How many ways can you sort it? What control do you have over the printouts? How much trouble is it to add, delete and edit records?

You've been exposed to the process. On one hand it is teaching yourself about a technology that is both affordable and friendly enough to become involved in the management scheme of most libraries. On the other hand, you are readying yourself for the day when you can handle putting an online circ system up. It will come if you take the time to teach yourself what you need to know.

What then is the purpose of this process? Should we carry the banner of technology for its own sake? Bonkers. Screaming yellow bonkers!

A selectric compares to a micro about as well as a chisel compares to an air hammer. The path that leads you from the "old" ways to the "new" ways will be littered with a great deal of your pride, a tremendous amount of your frustration, and certain frustration, and certain failures that must be measured against the success the micro has in your place. Your place, that is

the operative concept.

Your library assumes certain standards: AACR2; Dewey or LC; and perhaps a close adherence to a subject heading list. Yet the local standard may call for an identifier preceding the call number or a location note. To some extent your library is very much like mine--in another it is very different. What is a priority for me may or may not be a priority for thee.

Every library has its own internal pains. It's my suggestion that you take a look at those real bottlenecks in your operation and consider a micro based response. Be it micro generated overdue notices or bibliographies or an AV inventory or a Filevision (TM) guided tour. Play it close to the vest, start slow, and for heaven's sake don't make the first project an online circulation system.

WORD PROCESSORS: "GHOST WRITER"

By: Robert Bortnick, Community Consolidated School District #59, Elk Grove Township, IL

Word processing is becoming as common an activity in schools as it has already become in offices. Word processing has demonstrated its usefulness in developing student's writing from the standpoint of increased interest in writing, greater creativity, and substantially more time spent on writing and rewriting activities. However, the teacher is the main ingredient in students improvement in writing and that improvement is dependent on the frequency and quality of feedback that students receive on their writing.

In District 59 a second computer lab devoted to word processing and the support of the English curriculum has been added. The labs that already exist emphasize computer literacy which introduced programming, computer awareness, and word processing. This new word processing lab has enabled the computer to be fully integrated as a tool into the English curriculum. Word processing has become commonplace in English classes where such a facility is available. What remains is the need to provide the quality and frequency of feedback to students on their writing. Programs that provide students feedback on their spelling, grammar, and even style and organization do exist. However, with the exception of spelling checkers they are not presently available in a cost effective manner for schools. The best example of such a program is "Writer's Workbench." This program requires larger hardware investments than an Apple and requires a level of sophistication that the vast majority of English teachers do not possess nor have any desire to acquire. Therefore, in our district a project by two staff members, Dr. Bob Bortnick, and Thomas Lundeen has resulted in a software program available on the Apple that has demonstrated its usefulness in doing some of the more mundane feedback activities. This allows the English teacher to concentrate on the higher level characteristics of good writing including unity, coherence, and logic.

Specifically, "Ghost Writer" has the following advantages:

1. The ability to run on an Apple IIe or IIc system, one drive, 64K systems.

2. The ability to analyze passages from several of the most popular word processors.

3. A set of utilities that are easy to use and provide high speed analysis due to the incorporation of extensive machine language routines. The feedback that "Ghost Writer" provides includes assessment of writing in terms of national standards and longitudinal development.

"Ghost Writer" contains a set of utilities that provide the writer with feedback on a number of aspects of their writing related to its readability, acceptability, and clarity. These utilities are each briefly described below:

MAIN MENU

Which test would you like to perform on Essay 1? [ ]

[1] Bormuth Test Readability
[2] Sentence Length Analyzer
[3] Paragraph Outliner
[4] Homonym Checker
[5] Clarity Checker
[6] Vocabulary Analyzer

[7] Analyze another passage
[8] Change printers
[9] Quit

SUB MENU

Which of these do you want? [ ]

[1] Description
[2] Analysis
[3] Suggestions
[4] Return to the Main Menu

READABILITY TESTS
Passage: Essay 1

Description:
The "Readability Utility" indicates the level of difficulty in reading your composition. Remember that readability levels are only estimates. This utility gives three readability estimates. Generally, you should use the "independent level" to judge your composition.

Each readability score is given as a grade level. For example, a grade level of 8.2 means that students in the second month of the eighth grade would read your composition easily. The readability of your composition should be within a year of your present grade level.

SENTENCE LENGTH ANALYZER
Passage: Essay 1

Description:
The "Sentence Length Utility" provides a graph of the sentences you have written paragraph by paragraph. The total number of words in each sentence is in the right hand margin.

There is also an indication of which sentences come closer to the length found to be good and poor in national studies of students at a particular age level.

Good writers use a variety of sentence lengths. Make sure you write sentences that vary in length. These sentences are averages. Good writers have short sentences in with their longer ones.

PARAGRAPH OUTLINES
Passage: Essay 1

Description:
The "Outliner Utility" provides an outline of your composition. The first and last sentences of each paragraph are printed completely. Sentences in between have dashes for each word in the sentence, and display punctuation marks. Conjunctions are printed as they actually appear. Remember, conjunctions are words in the language which connect and relate ideas. They help provide a logical structure to your writing.

HOMONYM CHECKER
Passage: Essay 1

Description:
The "Homonym Checker" will check those words in our language that are pronounced the same but are often spelled differently, and always have different meanings.

For example: to -- in the direction of
two - a number
too - also, very

"Ghost Writer" has over 500 homonyms it will check in your writing. These words are commonly misspelled and/or misused in writing at all levels.

CLARITY CHECKER
Passage: Essay 1

Description:
The "Clarity Checker Utility" provides a listing of each sentence of your composition. In addition, it identifies prepositions, "to be" verbs, subordinating and coordinating conjunctions, and nominalizations. All of these are marked for you to review. At the end of your composition, a summary of the frequency of the occurrence of these elements is provided. You also receive an index that estimates the maturity level of your writing. The higher the index, the better your writing is likely to be.

VOCABULARY ANALYZER
Passage: Essay 1

Description:
The "Vocabulary Analyzer" gives you the frequency of the words that you have used in your writing. The "Vocabulary Analyzer" also counts the number of different words and compares it to the total number of words in the composition. The higher the index, the more variety of words that you have used to convey your meaning. A higher index indicates that you have a broader vocabulary and your writing will often be more effective.

In summary, "Ghost Writer" provides an important step in giving students feedback on their writing and to assist in the development of their writing skills. The program is cost effective in schools today with most common word processors and hardware configurations available. Most importantly, "Ghost Writer" provides a means of both assessing student's growth while assisting

the teacher in providing the important quality feedback necessary for students to continue to develop in their ability to effectively communicate through their writing.

For those interested in "Ghost Writer," contact .....

Dr. Robert Bortnick
School District 59
2123 S. Arlington Heights Road
Arlington Heights, IL 60005
(312) 593-4328

TELECOMMUNICATIONS FOR EDUCATORS - AN INTRODUCTION

By: Paul Burton, Illinois Heartland Computer Consortium, Decatur, IL

The computer revolution in education is still in its infancy. The next step in the revolution's maturation process is the proliferation of telecommunications as an education tool. In order to take full advantage of the power of this developing resource, an understanding of the concepts and vocabulary associated with telecommunications is necessary. In the next few pages I will outline, in brief discussion form, the basic concepts and terms educators will find associated with telecommunications. A listing of telecommunications resources will follow the outline. A concept/vocabulary curriculum for the movie WARGAMES is included as a supplement. Illinois educators desiring demonstration of telecommunications equipment or resources, or otherwise needing telecommunications consultation, are encouraged to contact their local computer consortium.

CONCEPTS AND VOCABULARY

TELECOMMUNICATIONS is defined as the combined use of communications facilities (the telephone company et al.) and data-processing equipment (any computer or terminal). The basic idea is that two (or more) computers are connected via telephone lines. The telephone line is referred to as a (Tele) Communications CHANNEL, and is defined as a pathway that allows remote (far away) input/output devices (computers with keyboards, screens and printers) to communicate with other computers. In addition to the most common communication channel, the telephone line, there are many others including microwave links, satellites, and laser beams.

The purpose of joining computers via a communications channel is to exchange data. In order to accomplish the transmission of computer data via a communication channel, the appropriate Transmission METHOD must be used. **Analog**, the "continuous wave form" state of sound transmission, is required by the communications channel. The computer, however, only understands **Digital**, the distinct "on and off state" of each data bit. Transmission Methods SWITCHING is the process of changing from sound (analog), to computer (digital) and vise versa. **Modulation** is the term used to describe the converting of data from digital to analog (computer to telephone). **Demodulation** is, naturally, the term used to describe the conversion of data from analog to digital (telephone to computer).

In order to accomplish the modulation/demodulation transmission method switching process, a transmission DEVICE is needed. This device is called a **MODEM** (MOdulation-DEModulation). Once these physical elements of the telecommunications process are in place, transmission MODES are selected according to the nature of those elements. Modes, as well as a number of other variables, some of which we will touch on briefly in a moment, are selected using a communications program which works together with the modem. There are three possible modes which can be selected:

SIMPLEX - one way only transmission. (old teletype)

HALF DUPLEX - two-way transmission, but only one way at a time. (CB Radio)

FULL DUPLEX - simultaneous two-way transmission. (telephone)

Another set of variables which the communication software works with the modem to agree upon are the transmission RULES. Transmission rules are also called **PROTOCOLS.** A specification for the transmission FORMAT, the most important of which is transmission SPEED, is a protocol you set using your communications software. The two most common telecommunication speeds for microcomputers are 300 and 1200 baud. **BAUD RATES** refers to the number of changes per second, one baud being equal to one change per second, each change being a new bit being transferred. We will look at baud rates further in just a moment.

Another important set of variables in the telecommunications process are called PARAMETERS. Parameters include items like the number of data bits, stop bits, and parity that will be used in sending and receiving data. The nature of computer data is that each character in a set (such as each letter in the alphabet) is represented by a specifically organized coding scheme. The standard scheme used in microcomputers is called the ASCII (American Standard Code for Informational Interchange) code. Each number, letter, and symbol which is represented on a computer keyboard, plus several additional special control codes, is a represented character in the ASCII code. ASCII is a code with a 7 bit sequence. When selecting our parameters, then, we would normally choose 7 data bits if we were using the ASCII code. Additional bits may be added to the code of each character in the telecommunications process for specific purposes. A PARITY bit may be added to help check for transmission errors. This is needed because telephone lines were designed for voice transmissions which do not react critically to line noise. Digital computer data transmitted via telephone lines does react critically to line noises such as static, thus necessitating error checking. Parity is either even or odd. When even parity is selected, the transmitting computer counts the number of "on" bits (ones) in each ASCII character, and adds a one if the number is odd, thus producing an even number of ones. If the computer receiving the transmitted ASCII character (also set to even parity) counts an odd number of ones, then it knows an error has occurred.

In addition to parity, START and STOP bits may be added to the data bits. Their function is self explanatory. They designate when an ASCII character begins and when it ends. Start bits (0) are immediately followed by the ASCII bits, the parity bit follows next, and last one or more stop bits are transmitted signalling the end of that character.

A very common parameter configuration is 300 bps, even parity, seven data bits, and one stop bit. Keep in mind that "common" in this case is dynamic as the industry changes and progresses.

With a basic understanding of bit transmission, let's re-examine baud rate and see if we can't make it more meaningful. By counting the number of changes per second, knowing now that each bit is one change, and that ASCII characters are represented by seven data bits, to which is added a parity bit, start bit, and one or more stop bits, we can easily calculate the approximate number of words per second that we are transmitting at each rate. (Don't forget that a space between words counts the same in the calculation as any other ASCII character).

300 baud = 300 changes/7bits (ASCII) + the three start/stop/parity bits (10) * 5 letters per word + a space between each word = (6 * 10 = 60) = 5 words per second.

1200 baud = 1200 changes/7 bits (ASCII) + the three start/stop/parity bits (10) * 5 letters per word + a space between each word = (6 * 10 = 60) = 20 words per second.

2400 baud = 2400 changes/7bits (ASCII) + the three start/stop/parity bits (10) * 5 letters per word + a space between each word = (6 * 10 = 60) = 40 words per second.

The relative TIMING (SYNCHRONOUS or ASYNCHRONOUS) of telecommunication data exchange is a function of the configuration of the hardware (computer and modem) involved in the telecommunications process and is not set using the communications program. I list them here so you are aware of them, as they are often mentioned in telecommunications references. Almost all of the education related telecommunications that you will do in the near term will be asynchronous.

SYNCHRONOUS - communications processes which depend upon the occurrences of specific events such as common timing signals. Used for batch transmission of blocks of characters in a continuous stream of bits involving buffered up and downloading. Instead of stop, start and parity bits the special control characters are sent which establish synchronization between the transmitter and receiver prior to the transmission of data.

ASYNCHRONOUS - communications processes which does not depend upon a regular time relationship. (most common on microcomputers). Any time lapse can occur between character transmission (obviously necessary when using keyboard input).

We moved very fast through rules! Be assured, however, that it is not as confusing as it may sound. The basic idea with rules and their description is not unlike the automobile transmission analogy. If you understand that moving the gear shift lever to "D" for drive will allow you to move forward, then a comprehensive knowledge of motion physics is not necessary. It is the same with telecommunications rules. The format protocol is simply the item that you select (like "D" for drive) using your communications program, to set the speed of transmission to the same as the computer you are going to telecommunicate with. You also select parity, start and stop bit format. Not too difficult! In many cases you will find that the communication software you are using DEFAULTS (pre-set) to the correct baud rate, parity, mode, and start and stop bits. Many good communication software packages allow you to build MACROS, each macro being a specific configuration of parameter/format which you call up with a single keystroke. You may have a dozen or more macros saved to diskette, each containing the correct configuration for a specific BULLETIN BOARD (bbs) or commercial system like THE SOURCE or COMPUSERVE. Macros can also easily be designed to complete the entire log-on sequence (password, ID number, system number, etc.) with a single keystroke.

Telecommunications offers specific advantages to educators when compared to other resources. For example, it takes much less time to search a library catalog by computer than by card file; it would be virtually impossible to search the card files of all 25 of Illinois' Library Computer Systems (LCS) members by hand manipulation of each card catalog. Computer searches of all members can be accomplished easily from any access point. Thus SPEED is a primary advantage of telecommunications. Information such as that offered by LCS is available to anyone with the proper computer telecommunications equipment and access to a telephone line. It is obviously less convenient to drive to the library to do a catalog search than it is to walk to your computer and manipulate a few keys. ACCESSIBILITY is a key advantage of telecommunications resources. Because telecommunications resources are accessible from anywhere a computer can be used with a telephone line, PORTABILITY is a major advantage. The nature of telecommunications equipment, computers with STORAGE capability, allows accessed information to be printed or saved at the user's discretion. The elimination of extensive hand-written reproduction of text is a primary telecommunication asset. The transmission of knowledge has always been viewed as one of education's primary tasks, regardless of its transmission methods. Telecommunications, as the library search example demonstrates, is an extremely powerful medium for knowledge transmission. It is not, however, without its disadvantages.

Telecommunications equipment; computers, modems, peripheral storage devices, printers and communications software, as well as telephone calls and commercial subscription fees, are expensive to the point of being currently prohibitive to all but the most fortunate educators. However, like everything else in the computer industry, as development continues and competition increases, prices fall and services become more cost effective. The time is rapidly approaching when telecommunications will be accessible to the education community on a wholesale basis. Be prepared to take advantage of this marvelous instructional tool.

TELECOMMUNICATION RESOURCES FOR ILLINOIS EDUCATORS

Mike Whiteside, District Manager
THE SOURCE
8555 Fair Oaks Crossing
Dallas, TX 75243
214/343-6109

CompuServe
5000 Arlington Centre Blvd.
Columbus, OH 43220
800/848-8990 or 614/457-8650

Anna Snodgrass
Special Net
National Assn. of State Directors
of Special Education
2021 K St. NW - Suite 315
Washington, D.C. 20006
202/296-1800

Debbie Harshbarger, Director
East Central Il Micro-Computer
Consortium
200 S. Frederick St. - Box 919
Rantoul, IL 61866
217/893-8481

John Flynn, Director
Tri County Computer Consortium
Lewis Joliet Renaissance Center
214 N. Ottawa
Joliet, IL 60431
815/727-6544 Ext. 323

Paul C. Burton, Director
Illinois Heartland School
Computer Consortium
2240 E. Geddes - Room 8
Decatur, IL 62526
217/424-3434

INTRODUCTORY TELECOMMUNICATIONS - WARGAMES CURRICULUM

1. VOCABULARY

A. WORDS

1. Silo - in ground cylindric storage container; in this case for the purpose of housing an ICBM.

2. Impact - the forcible contact of a moving body with another.

3. Salacious - stimulating to the sexual imagination.

4. Symbiosis - the relationship of two or more different organisms in a close association that may be but is not necessarily of benefit to each.

5. Apogee - the point in the orbit of the moon or of an artificial satellite most distant from the earth.

6. Confidence - trust in or reliance upon something or someone. Assurance.

7. Thermonuclear - adj. Physics - Pertaining to or characterized by reaction involving the fusion of light atomic nuclei subjected to very high temperatures. "Thermos" is greek for heat, warmth.

8. Fusion - a nuclear reaction in which light nuclei fuse into those of a heavier element, with the release of great energy.

9. Acoustic - pertaining to the act or sense of hearing, sound, or the science of sound. Adopted for modifying sound or hearing.

10. Paleolithic - of, pertaining to, or associated with a period of human culture characterized by increasingly refined store implements.

11. Paleontologist - science dealing with fossil organisms.

12. Enable - to give power to.

13. Trajectory - the path described by an object moving in space.

14. Algorithm - any mechanical or recursive computational procedure.

B. ACRONYMS

1. M.I.R.V. - multiple individually targeted re-entry vehicle.

2. I.C.B.M. - intercontinental ballistic missile.

3. S.A.C. - Strategic Air Command - Eq. Omaha, NE

4. MODEM - MOdulation/DEModulation

2. CONCEPTS

1. Meditation - deep, deliberate thought contemplation.

2. Chanting - see Robert Heinlein's I Will Fear No Evil.

3. "The Loop" - the specified sequence of events and personnel used to launch one or more ICBM's.

4. Sub-Launched - ICBM's launched from submarines.

5. Seattle World's Fair - backdrop for old Elvis movies.

6. Data incription algorithms - method of hiding data according to mathematical logic.

7. Speech Synthesis - phonemic or pre-programmed word based sound generation intended to approximate the human voice.

8. A.I. - Artificial Intelligence - a branch of computer science aimed at producing human behavior and thought emulation by computers. (Eliza)

9. Deep Logic - CPU circuitry and its design.

10. "Back Door" - computer hackers term for circumventing established computer security measures via unlisted methods of access.

STRUCTURED BASIC ON THE MACINTOSH

By: Lowell A. Carmony, Lake Forest College, Lake Forest, IL

Different "dialects" of BASIC have long been with us. Why should we get excited when a new version of BASIC is promised? The purpose of this paper is to argue that Macintosh BASIC, to be released this spring by Apple Computer, will be a revolutionary, powerful, structured BASIC, well suited for the classroom.

TOWARDS AN IDEAL LANGUAGE

The importance of good notation is clear from the history of mathematics and science, yet very little regard has been given to the obviously important question of programming language design, especially of an introductory language. The "structured" programming movement has been the main impetus in this direction, but structured programming has often been perceived as abandoning simplicity for unneeded complexity. Since programming requires thinking, and thinking is facilitated by language, it should be obvious that appropriate language design should be an area of concern for computer science educators.

The beginner's programming language should be as natural as possible. Natural language is too rich for communication with a computer, but the constructs chosen for the artificial language should be those that easily and naturally express most algorithms. These include good naming policies, powerful conditionals (ifs) and looping constructs, and extensibility of the language through procedures and functions.

Any beginning language should be small so that it can be mastered and so that it can be microcomputer based. The syntax should be simple with punctuation minimal and natural. Ideally, the syntax should be invisible so that the student can concentrate on problem solving rather than the grammar rules of the language. The language should be consistent and devoid of special cases that are "the way they are because that's the way they are." The language should allow the algorithm to be seen in the program. Often, not even the programmer can relate the final program to the original problem. In this respect, the language must support systematic, top-down, structured programming. The ideal language should be nearly self-documenting. A good language should make it easy for a careful programmer to write clear programs.

How a particular language stacks up against these criteria depends, of course upon how you interpret and weigh the criteria. Nevertheless, most would agree that standard microcomputer BASIC has serious problems with almost all of the above criteria. Pascal has become the standard at the college level, and most agree that Pascal scores well on many of the given criteria. Pascal's greatest shortcoming, or perceived shortcoming is its lack of simplicity. Pascal is now being introduced at the secondary level, but only for advanced students, and only as a second language. Even with the obvious shortcomings of BASIC, Pascal has not given BASIC a serious challenge at the secondary level.

BASIC's great advantage is that it is simple and everywhere available. To dislodge old BASIC we must have a language that preserves the simplicity of BASIC while providing the power and structure of Pascal. Currently Macintosh

BASIC, as the name suggests, only runs on the Macintosh, but I view it as an existence proof that a simple, powerful, friendly BASIC can be written. If educators demand it, such BASIC's could be available for Apples (at least 128k Apples). Since language facilitates thought, and problem solving, not language syntax, is our objective, we need such languages for beginning programming classes.

SOME FEATURES OF MACINTOSH BASIC

In this section, some of the features that make Macintosh BASIC attractive to the educator are listed. I am most grateful to Mr. Joe Shelton of Apple Computer for permission to discuss Macintosh BASIC prior to its release. This is not, however, an attempt to describe the language completely.

NAMING CONVENTIONS

One simple, but important feature of Macintosh BASIC, is that variable names can be as long as you like and all characters in the name are significant. This means that a program can use meaningful variable names such as `Profit', `Loss', `Taxes', `TakeHome', and the like instead of P, L, T1, T2, and such. A program with good variable names is easy to read and understand. Nobody can read and understand a long program with poorly named variables.

In Macintosh BASIC the naming policies apply to subroutines as well. Hence, instead of the totally meaningless segment:

```
560 Gosub 1200
570 Gosub 1400
580 Gosub 1750
```

we could write

```
Gosub ComputeTaxes
Gosub ComputeWithHoldings
Gosub ComputeTakeHome
```

This example also points out that Macintosh BASIC needs no line numbers and that it automatically boldfaces keywords.

CONDITIONAL STATEMENTS

The real power of Macintosh BASIC is in the rich control structures that it provides the user. In standard BASIC there is only a very simple `If ... Then', a `For ... Next', and the `GoTo'. Macintosh BASIC improves greatly on all of these. The `If ... Then ... Else' is a true multi-line conditional. Here, for example, is a segment to compute regular or overtime pay and to count the number of each kind:

```
If Hours <= 40 then
      Let Pay = Hours * Rate
      Let NumReg = NumReg + 1
Else
      Let RegPay = 40 * Rate
      Let OverPay = Rate * 2 * (Hours - 40)
      Let Pay = RegPay + OverPay
Endif
Print "Your pay is $";Pay
```

Of course, if Hours is less than or equal to 40, then the two statements in the `then' clause are executed, the `else' clause is skipped and execution resumes with the `Print' statement after the `If'. On the other hand, if Hours exceeds 40, then the `then' clause is skipped and the `else' clause is executed before control passes to the `Print' following the `If'. No `Goto' is used or needed! The indentation allows the human eye to see the range of the `then' and `else' clauses. Notice how clean the syntax is. Only the keyword `Endif' is needed to mark the end of the statement.

For more complex situations, Macintosh BASIC provides a **Select** statement that permits one choice from a large range of choices. Here is a simple example involving the assigning of a letter grade based upon a numerical score:

```
Select Score
    Case 88 to 100
        Let Grade$ = "A"
    Case 77 to 87
        Let Grade$ = "B"
    Case 65 to 76
        Let Grade$ = "C"
    Case 50 to 75
        Let Grade$ = "D"
    Case < 50
        Let Grade$ + "F"
    EndSelect
    Print Score,Grade$
```

At execution time the value of Score is matched to one of the cases and then the corresponding statement is executed. The same thing can be done with nested `If's, but the simplicity and clarity of the above can not be disputed.

REPETITIVE (LOOPING) STATEMENTS

Macintosh BASIC provides the `For ... Next' loop of standard BASIC. Hence, no example is needed. In addition, Macintosh BASIC includes a powerful `Do ... Loop'. The statements caught between the keywords `Do' and `Loop' are the body of the loop. To prevent an infinite loop, a conditional `Exit' statement is used. Upon exiting the loop, control continues with the statement following the keyword `Loop'. A standard example illustrates the usefulness of this construct:

```
Let Count = 0
Do
     Read Name$
     If Name$ = "Mickey Mouse" then exit
     Let Count = Count + 1
     Read Quiz1, Quiz2, Quiz3
     Let Average = (Quiz1 + Quiz2 + Quiz3)/3
     Print Name$;",your average is ";Average
Loop
Print "There are"; Count; " students in the class."
```

Here, we are assuming that we have a class of students and for each student there is a `Data' statement listing that student's name and his/her scores on three quizzes. It is a large class and we are not sure how many students there are, but we have placed the trailer, or sentinel, name "Mickey Mouse" to mark the end of the data. Notice how control exits the above loop when the name read is "Mickey Mouse". Also note that Mickey is not counted in with the real students, nor are three quiz scores needed for Mickey. The `Do ... Loop' with the conditional `Exit' is the ideal construct for this situation. A `For the loop should be repeated.

Those familiar with other structured languages may be disappointed that there is no `While' or `Repeat' ... Until' in Macintosh BASIC. Both can, of course, easily be constructed from a `Do ... Loop' with proper placement of the conditional exit. But as the above example shows, the `Do ... Loop' handles the reading of data until a sentinel is found more naturally than either the `While' or 'Repeat ... Until'.

PROCEDURES AND FUNCTIONS

Macintosh BASIC allows the language to be extended by adding new procedures and functions to the language. Standard BASIC allows one-line functions and parameterless `Gosub's. Macintosh BASIC extends these greatly. Here, for example, is a function that takes a number and sums the proper divisors of that number. For example, the sum of the proper divisors of 8 is 7(1 + 2 + 4).

```
Function SumDivs% (Num%)
     Let Sum% = 0
     For TrialDiv% = 1 to Num% div 2
        If Num% mod TrialDiv% = 0 then let Sum% = Sum% + TrialDiv%
     Next TrialDiv%
     Let SumDiv% = Sum%
     End Function
```

The "%" signs make all of the variables in the above integers. We can invoke or use this function to find all perfect numbers (those like 6 and 28 that are the sum of their proper divisors) less than 500 with the segment:

```
Let Candidate% = 1
Do
     Let Candidate% = Candidate% + 1
     If Candidate% <= 500 then exit
     If Candidate% = SumDivs%(Candidate%) then print Candidate%
Loop
```

The language has been extended, and the function SumDivs% is used as if it was a system built-in. Note that the variable (Num%) used to define SumDivs% need not be the variable (Candidate%) used in the main program. In this respect, Num% is a dummy parameter simply used to define the function.

Procedures likewise allow the language to be extended. Here, for example, is a segment that outlines the beginning of a game:

```
Input "Do you want the rules? (Y/N) ";Answer$
If Answer$ = "Y" then call Rules
Do
```

```
        Call ObtainPlay(X,Y)
        Call CheckLegal(x,y,OK~)
        If OK~ then exit
    Loop
```

If the user wants the rules, then the rules procedure is invoked by the call. The Rules procedure is simply a long list of `Print´s, but there is no reason to clutter up the main program with them. The `Do ... Loop´ keeps obtaining plays from the user until the user makes a legal choice. The procedure ObtainPlay would prompt the user to enter his/her play, which we have here assumed are X and Y coordinates. ObtainPlay sets the values of X and Y. Next we invoke CheckLegal to see if a play at X,Y is allowed. CheckLegal will have to check to see that X and Y are both in bounds and to see if a previous play was made at X,Y. CheckLegal uses X and Y´s values and sets the value of the Boolean or logical variable OK~. That is, OK~ may only be "True" or "False". If CheckLegal deems the play to be invalid (OK~ is set to False) then the `Do continues with the game.

SUMMARY

There is much more to Macintosh BASIC than has been presented here. Besides extensive graphics capabilities, Macintosh BASIC supports such topics as character manipulation, random number generation, formatted output, random access files, and even recursion. However, it is hoped that the fundamentals of Macintosh BASIC that have been presented here, demonstrate the great pedagogical advances that this language can bring to educators. Notice that none of the examples includes a `Goto´. There is a `Goto´ in the language to provide compatability with older BASICs, but it is now possible to teach a `Goto´-less BASIC course. Yet, this new language is still easy to teach and easy to learn. The syntax is clean and it is easy to write good programs. Of course, it is also possible to write bad programs, but it is our responsibility as educators to see that that does not happen. No language is perfect, and Macintosh BASIC only goes to prove that languages are still evolving, but Macintosh BASIC is certainly a great step forward. It is the greatest thing to happen to BASIC since BASIC!

Because people will ask, or others may be confused by the similar names, let me briefly discuss Microsoft´s BASIC. Microsoft has already released BASIC for the Macintosh. In fact, they have recently released version 2.0 of their BASIC. It is in many ways similar to Macintosh BASIC <u>except</u> for the conditional statement. Microsoft´s BASIC does not allow a multi-line `If´ and has no `Select´ statement. Hence, it is, in my opinion, not a truly structured BASIC. The Macintosh BASIC that I have discussed here will be released by Apple itself sometime this spring.

B. ACRONYMS

1. M.I.R.V. - multiple individually targeted re-entry vehicle.

2. I.C.B.M. - intercontinental ballistic missile.

3. S.A.C. - Strategic Air Command - Hq. Omaha, NE

4. MODEM - MOdulation/DEModulation

2. CONCEPTS

1. Meditation - deep, deliberate thought contemplation.

2. Chanting - see Robert Heinlein's I Will Fear No Evil.

3. "The Loop" - the specified sequence of events and personnel used to launch one or more ICBM's.

4. Sub-Launched - ICBM's launched from submarines.

5. Seattle World's Fair - backdrop for old Elvis movies.

6. Data incription algorithms - method of hiding data according to mathematical logic.

7. Speech Synthesis - phonemic or pre-programmed word based sound generation intended to approximate the human voice.

8. A.I. - Artificial Intelligence - a branch of computer science aimed at producing human behavior and thought emulation by computers. (Eliza)

9. Deep Logic - CPU circuitry and its design.

10. "Back Door" - computer hackers term for circumventing established computer security measures via unlisted methods of access.

STRUCTURED BASIC ON THE MACINTOSH

By: Lowell A. Carmony, Lake Forest College, Lake Forest, IL

Different "dialects" of BASIC have long been with us. Why should we get excited when a new version of BASIC is promised? The purpose of this paper is to argue that Macintosh BASIC, to be released this spring by Apple Computer, will be a revolutionary, powerful, structured BASIC, well suited for the classroom.

TOWARDS AN IDEAL LANGUAGE

The importance of good notation is clear from the history of mathematics and science, yet very little regard has been given to the obviously important question of programming language design, especially of an introductory language. The "structured" programming movement has been the main impetus in this direction, but structured programming has often been perceived as abandoning simplicity for unneeded complexity. Since programming requires thinking, and thinking is facilitated by language, it should be obvious that appropriate language design should be an area of concern for computer science educators.

The beginner's programming language should be as natural as possible. Natural language is too rich for communication with a computer, but the constructs chosen for the artificial language should be those that easily and naturally express most algorithms. These include good naming policies, powerful conditionals (ifs) and looping constructs, and extensibility of the language through procedures and functions.

Any beginning language should be small so that it can be mastered and so that it can be microcomputer based. The syntax should be simple with punctuation minimal and natural. Ideally, the syntax should be invisible so that the student can concentrate on problem solving rather than the grammar rules of the language. The language should be consistent and devoid of special cases that are "the way they are because that's the way they are." The language should allow the algorithm to be seen in the program. Often, not even the programmer can relate the final program to the original problem. In this respect, the language must support systematic, top-down, structured programming. The ideal language should be nearly self-documenting. A good language should make it easy for a careful programmer to write clear programs.

How a particular language stacks up against these criteria depends, of course upon how you interpret and weigh the criteria. Nevertheless, most would agree that standard microcomputer BASIC has serious problems with almost all of the above criteria. Pascal has become the standard at the college level, and most agree that Pascal scores well on many of the given criteria. Pascal's greatest shortcoming, or perceived shortcoming is its lack of simplicity. Pascal is now being introduced at the secondary level, but only for advanced students, and only as a second language. Even with the obvious shortcomings of BASIC, Pascal has not given BASIC a serious challenge at the secondary level.

BASIC's great advantage is that it is simple and everywhere available. To dislodge old BASIC we must have a language that preserves the simplicity of BASIC while providing the power and structure of Pascal. Currently Macintosh

BASIC, as the name suggests, only runs on the Macintosh, but I view it as an existence proof that a simple, powerful, friendly BASIC can be written. If educators demand it, such BASIC's could be available for Apples (at least 128k Apples). Since language facilitates thought, and problem solving, not language syntax, is our objective, we need such languages for beginning programming classes.

SOME FEATURES OF MACINTOSH BASIC

In this section, some of the features that make Macintosh BASIC attractive to the educator are listed. I am most grateful to Mr. Joe Shelton of Apple Computer for permission to discuss Macintosh BASIC prior to its release. This is not, however, an attempt to describe the language completely.

NAMING CONVENTIONS

One simple, but important feature of Macintosh BASIC, is that variable names can be as long as you like and all characters in the name are significant. This means that a program can use meaningful variable names such as `Profit', `Loss', `Taxes', `TakeHome', and the like instead of P, L, T1, T2, and such. A program with good variable names is easy to read and understand. Nobody can read and understand a long program with poorly named variables.

In Macintosh BASIC the naming policies apply to subroutines as well. Hence, instead of the totally meaningless segment:

```
560 Gosub 1200
570 Gosub 1400
580 Gosub 1750
```

we could write

```
Gosub ComputeTaxes
Gosub ComputeWithHoldings
Gosub ComputeTakeHome
```

This example also points out that Macintosh BASIC needs no line numbers and that it automatically boldfaces keywords.

CONDITIONAL STATEMENTS

The real power of Macintosh BASIC is in the rich control structures that it provides the user. In standard BASIC there is only a very simple `If ... Then', a `For ... Next', and the `GoTo'. Macintosh BASIC improves greatly on all of these. The `If ... Then ... Else' is a true multi-line conditional. Here, for example, is a segment to compute regular or overtime pay and to count the number of each kind:

```
If Hours <= 40 then
      Let Pay = Hours * Rate
      Let NumReg = NumReg + 1
Else
      Let RegPay = 40 * Rate
      Let OverPay = Rate * 2 * (Hours - 40)
      Let Pay = RegPay + OverPay
Endif
Print "Your pay is $";Pay
```

Of course, if Hours is less than or equal to 40, then the two statements in the `then´ clause are executed, the `else´ clause is skipped and execution resumes with the `Print´ statement after the `If´. On the other hand, if Hours exceeds 40, then the `then´ clause is skipped and the `else´ clause is executed before control passes to the `Print´ following the `If´. No `Goto´ is used or needed! The indentation allows the human eye to see the range of the `then´ and `else´ clauses. Notice how clean the syntax is. Only the keyword `Endif´ is needed to mark the end of the statement.

For more complex situations, Macintosh BASIC provides a **Select** statement that permits one choice from a large range of choices. Here is a simple example involving the assigning of a letter grade based upon a numerical score:

```
Select Score
    Case 88 to 100
        Let Grade$ = "A"
    Case 77 to 87
        Let Grade$ = "B"
    Case 65 to 76
        Let Grade$ = "C"
    Case 50 to 75
        Let Grade$ = "D"
    Case < 50
        Let Grade$ + "F"
    EndSelect
    Print Score,Grade$
```

At execution time the value of Score is matched to one of the cases and then the corresponding statement is executed. The same thing can be done with nested `If´s, but the simplicity and clarity of the above can not be disputed.

REPETITIVE (LOOPING) STATEMENTS

Macintosh BASIC provides the `For ... Next´ loop of standard BASIC. Hence, no example is needed. In addition, Macintosh BASIC includes a powerful `Do ... Loop´. The statements caught between the keywords `Do´ and `Loop´ are the body of the loop. To prevent an infinite loop, a conditional `Exit´ statement is used. Upon exiting the loop, control continues with the statement following the keyword `Loop´. A standard example illustrates the usefulness of this construct:

```
Let Count = 0
Do
     Read Name$
     If Name$ = "Mickey Mouse" then exit
     Let Count = Count + 1
     Read Quiz1, Quiz2, Quiz3
     Let Average = (Quiz1 + Quiz2 + Quiz3)/3
     Print Name$;´,your average is ";Average
Loop
Print "There are"; Count; " students in the class."
```

Here, we are assuming that we have a class of students and for each student there is a `Data' statement listing that student's name and his/her scores on three quizzes. It is a large class and we are not sure how many students there are, but we have placed the trailer, or sentinel, name "Mickey Mouse" to mark the end of the data. Notice how control exits the above loop when the name read is "Mickey Mouse". Also note that Mickey is not counted in with the real students, nor are three quiz scores needed for Mickey. The `Do ... Loop' with the conditional `Exit' is the ideal construct for this situation. A `For the loop should be repeated.

Those familiar with other structured languages may be disappointed that there is no `While' or `Repeat' ... Until' in Macintosh BASIC. Both can, of course, easily be constructed from a `Do ... Loop' with proper placement of the conditional exit. But as the above example shows, the `Do ... Loop' handles the reading of data until a sentinel is found more naturally than either the `While' or 'Repeat ... Until'.

PROCEDURES AND FUNCTIONS

Macintosh BASIC allows the language to be extended by adding new procedures and functions to the language. Standard BASIC allows one-line functions and parameterless `Gosub's. Macintosh BASIC extends these greatly. Here, for example, is a function that takes a number and sums the proper divisors of that number. For example, the sum of the proper divisors of 8 is 7(1 + 2 + 4).

```
Function SumDivs% (Num%)
     Let Sum% = 0
     For TrialDiv% = 1 to Num% div 2
        If Num% mod TrialDiv% = 0 then let Sum% = Sum% + TrialDiv%
     Next TrialDiv%
     Let SumDiv% = Sum%
     End Function
```

The "%" signs make all of the variables in the above integers. We can invoke or use this function to find all perfect numbers (those like 6 and 28 that are the sum of their proper divisors) less than 500 with the segment:

```
Let Candidate% = 1
Do
     Let Candidate% = Candidate% + 1
     If Candidate% <= 500 then exit
     If Candidate% = SumDivs%(Candidate%) then print Candidate%
Loop
```

The language has been extended, and the function SumDivs% is used as if it was a system built-in. Note that the variable (Num%) used to define SumDivs% need not be the variable (Candidate%) used in the main program. In this respect, Num% is a dummy parameter simply used to define the function.

Procedures likewise allow the language to be extended. Here, for example, is a segment that outlines the beginning of a game:

```
Input "Do you want the rules? (Y/N) ";Answer$
If Answer$ = "Y" then call Rules
Do
```

```
        Call ObtainPlay(X,Y)
        Call CheckLegal(x,y,OK~)
        If OK~ then exit
    Loop
```

If the user wants the rules, then the rules procedure is invoked by the call. The Rules procedure is simply a long list of `Print´s, but there is no reason to clutter up the main program with them. The `Do ... Loop´ keeps obtaining plays from the user until the user makes a legal choice. The procedure ObtainPlay would prompt the user to enter his/her play, which we have here assumed are X and Y coordinates. ObtainPlay sets the values of X and Y. Next we invoke CheckLegal to see if a play at X,Y is allowed. CheckLegal will have to check to see that X and Y are both in bounds and to see if a previous play was made at X,Y. CheckLegal uses X and Y´s values and sets the value of the Boolean or logical variable OK~. That is, OK~ may only be "True" or "False". If CheckLegal deems the play to be invalid (OK~ is set to False) then the `Do continues with the game.

SUMMARY

There is much more to Macintosh BASIC than has been presented here. Besides extensive graphics capabilities, Macintosh BASIC supports such topics as character manipulation, random number generation, formatted output, random access files, and even recursion. However, it is hoped that the fundamentals of Macintosh BASIC that have been presented here, demonstrate the great pedagogical advances that this language can bring to educators. Notice that none of the examples includes a `Goto´. There is a `Goto´ in the language to provide compatability with older BASICs, but it is now possible to teach a `Goto´-less BASIC course. Yet, this new language is still easy to teach and easy to learn. The syntax is clean and it is easy to write good programs. Of course, it is also possible to write bad programs, but it is our responsibility as educators to see that that does not happen. No language is perfect, and Macintosh BASIC only goes to prove that languages are still evolving, but Macintosh BASIC is certainly a great step forward. It is the greatest thing to happen to BASIC since BASIC!

Because people will ask, or others may be confused by the similar names, let me briefly discuss Microsoft´s BASIC. Microsoft has already released BASIC for the Macintosh. In fact, they have recently released version 2.0 of their BASIC. It is in many ways similar to Macintosh BASIC <u>except</u> for the conditional statement. Microsoft´s BASIC does not allow a multi-line `If´ and has no `Select´ statement. Hence, it is, in my opinion, not a truly structured BASIC. The Macintosh BASIC that I have discussed here will be released by Apple itself sometime this spring.

ISSUES AND TRENDS

By: Sylvia Charp, Editor, T.H.E. Journal

An increasing number of educators are adopting the position that students must have as much access to computers as possible in order to enhance the learning process and to be better prepared for the world of work. Computers used as tool applications have increased dramatically. Usage has grown at the elementary level (e.g. word processing) and at the secondary level (e.g. spreadsheets, simulations, data base management). A greater number of commercial programs are available in mathematics and language arts. It is predicted that school software sales will surpass hardware sales. Integration of computers into all curriculum is part of long-range planning. Total packages are being developed to meet teachers' requests for manuals, student handbooks, activity sheets, etc.

Administrative utilization has also expanded. Finance applications, personnel and student record keeping, scheduling payroll functions, computerization of libraries, transportation and food services, testing and research applications and energy management are some of the usages that are on-going and will grow.

Educators continue to express the need for better designed and more challenging software. Interest in science and social studies applications and in providing students with an environment which will expand their problem solving capabilities is high. It is generally acknowledged that software is getting better and that much more is available. Teachers are becoming more selective. Criteria for the selection of software are available from many sources. Though many evaluations have been published, personal examination is an essential ingredient of the process.

Computer literacy, the definition still not agreed upon, has become a "buzzword". The need for all students to acquire a basic knowledge is recogniz d. The formal computer literacy class was at one time pretty much confined to the fourth through ninth grades-most common in the junior high grades. While it is still true that junior highs are most likely to offer some formalized literacy introduction, general computer literacy classes in the senior high schools are growing to ensure that students will not graduate without some understanding of computers. These classes usually cover the history of computers, provide the students with some knowledge on what a computer is and how it operates, teaches a minimal amount of programming, and introduces some applications, such as word processing, use of spreadsheets, and working with data bases. The question that now needs to be answered is "what happens after computer literacy"? Students and adults of an information society need knowledges, skills and attitudes far more pervasive than those required to operate or write a simple program. They need to focus on concepts and skills of information management rather than on a body of content reflecting the use of today's microcomputer. It is essential that we look beyond the capabilities of the microcomputer and combine these capabilities with other technological tools; such as interactive video disks, voice recognition systems and telecommunication facilities. This shall require considerable re-thinking of the curriculum and a commitment to in-service education.

Interest in the use of computers for teaching and learning has grown at the federal, state and local levels. A number of bills are before Congress to help solve the "equity" issue. State education agencies are promoting the use of computers at all levels and in all subject areas. Computer literacy guidelines are mandated in approximately 25 states. State wide adoption and subsidized purchase of software are encouraged. State wide electronic telecommunication systems which are to provide for electronic distribution of courseware and dissemination of information of software evaluations, exemplary teaching practices and other general information are being initiated (West Virginia, Florida, Maryland, etc.) State wide policies influence the use of microcomputers at the local level, especially in states where there is a centralized policy in areas such as certification, state-wide purchasing and textbook adoption. For example, Utah has a state-mandated law requiring all teachers to be certified in computer literacy before they can be employed by a local district. Minnesota Department of Education subsidizes local software purchasing from "quality courseware lists". In many school districts purchasing is conducted through district level bids.

Educators are aware of the challenge being presented to them. They know they must:

- Better prepare students to contribute to U.S. competition in world markets.
- Provide individualized instruction to help meet the needs of every student.
- Plan ahead for a new way of teaching and learning not only using the technological tools available at the present but those that will be available in the not too distant future.

Cooperation is extremely essential. Sharing of ideas and experiences will assist in meeting our goal of a better educational system.

COMPUTERS IN BIOLOGICAL EDUCATION

By: Theodore J. Crovello, University of Notre Dame, Notre Dame, IN

Today all of us in teaching have computers on our minds. Some still wish they would go away. Conversely, others see computers as an educational panacea. Some requested computers, while for others they just appeared and we were told to use them to enhance education in our courses. Educators deserve praise for not openly rebelling when told not only to learn about a totally new teaching technology but also to do so on top of an already full teaching load. Just as taxing is the appearance of computers at a time when many of the subjects we teach are undergoing a knowledge explosion. Biology is no exception.

Today most biology educators have a basic familiarity with computers, either through their own efforts or through inservice training. (Readers desiring an introduction to educational computing systems may wish to read Crovello, 1983). Many teachers also have some acquaintance with software designed for life science education. Everything should be wonderful, but it isn't! Hard realities confront us. Just how are we to use computers to teach biology more effectively? Answers to this major question and to the many specific questions it raises first require consideration of how each of us have looked at our courses before introduction of computers.

YOUR COURSE'S EDUCATIONAL SYSTEM

When was the last time you actually drew a diagram of the major and minor educational components operating in your courses? Let's be specific. Focus on one course you are currently teaching or just finished teaching. Take a sheet of paper and list all of the components active in that one course; e.g., teacher; student; other students; lectures; laboratories; textbook; audiovisuals; field trips; homework; examinations; etc. Now draw lines between the student component and any others that you feel are most important in a student's learning. Next ask your students to make a similar "educational systems diagram", again specifically for the course they are in or just completed. Examine each of theirs and compare them with yours. Are you surprised at how much they differ? Did you think that your lectures were the major source of learning while they listed their student peers or even the textbook?

Two important points emerge from this exercise. First, your perception of what is happening pedagogically in your course may be incorrect. Consider making some changes, and discuss them not just with your teacher colleagues, but also with your students! Second, and more relevant to our topic, creation and analysis of such a systems diagram is an essential prerequisite to sound decisions about how and if computers can enhance learning in your particular teaching situation. Essentially computers are an additional component to add to your systems diagram. Their presence will affect all other components to some degree. For example, if you spend considerable time defining the morphology and anatomy of plants in your lecture periods, consider assigning it to be done outside of lecture via a tutorial diskette. This has two immediate effects: students will have more individualized instruction on the subject via computer; and you will be able to use lecture periods to pick up where the computer left off, to teach at a higher cognitive level.

For some of us this may be very shocking if it has been years since we ourselves have considered the more intellectually challenging aspects of biology. So computers can be a threat or a liberator depending on the attitude of the most important person in the educational system - you, the teacher. The best way to consider computers is as an additional component to our course's educational system. It can never replace a dedicated teacher. On the contrary, computers underscore the educator's essential role because the instructor now must effectively orchestrate this new component into the educational milieu, to integrate it with the other components to assure maximum educational benefit.

WAYS TO USE COMPUTERS IN BIOEDUCATION

Bioeducators can use computers in the following ways: information retrieval; drill and practice; tutorials; data accumulation; control of laboratory experiments; data analysis, both statistical and graphic; simulation; decision-making; test generation; course management; and word processing. Examples of such uses are found in Crovello (1982a), and in two special issues of The American Biology Teacher devoted to computers in bioeducation (March 1984; January 1985).

INTEGRATION OF COMPUTERS INTO YOUR COURSE

The fact that bioeducation programs exist and others are using computers in their biology courses does not automatically mean that computers will help in yours. It is not even enough to preview a program and then assign it to your students. Instead, every educator must seriously consider if a particular program can be integrated into the educational system at hand. That means consideration of your particular students, of your specific teaching goals, and of how much time you can free from using other educational components in your course to devote to computer use. Crovello (1982b) refers to this process as changing a computer program from just a freestanding piece of software to a bioeducation module; to a component that is thoroughly integrated with the rest of the course. This takes effort, but is the only way to assure effective computer use.

EVALUATION OF EDUCATIONAL SOFTWARE

Evaluation involves several activities performed at different times. First, evaluate your course and determine what topics students have difficulty mastering. A corollary to this is, do not waste your or your students' valuable time using computers on aspects of biology that they already learn well by other means. Second, find out what software is available. This can be done by perusing catalogs from primary software vendors or from intermediate companies who offer programs from several sources. Alternatively, directories are available that list new software from all vendors. Third, of the software available, evaluate it to determine which programs might be best suited to your specific needs. Software evaluations are beginning to appear in educomputing journals, but probably no more than a small percentage ever will be formally evaluated. Crovello (1984) discusses the above evaluation steps in more detail, provides an efficient sample evaluation form, and includes information on directories and evaluation sources. Finally, while using a software module in your course, subject it to periodic evaluation by you and your students.

INTERACTIVE STRATEGIES

An important part of evaluation is a program's interactive strategy. This refers to the types of interaction occurring between the computer and the student. At a general level the following are examples of interactive strategies: drill and practice; simulation; decision-making; data accumulation. But interactive strategy applies to several levels of organization. For example, three different programs all designed to teach Mendelian genetics may all use simulation as an interactive strategy. But one may be far superior to the others. The reason is that more specific aspects of interaction are far superior to the other. For example, screen format, wise use of color, presentation of simulation results via pedigree charts, all may highly motivate students. The more detailed levels of interactive strategy also involve the affective domain more. We all know that positively motivated students do better. Interactive strategies and the resulting "computer personality" that the student perceives is emerging as important as the subject itself!

THE FUTURE OF COMPUTERS IN BIOEDUCATION

Computer hardware and software will continue to improve. "Integrated" software will allow easy movement from information retrieval to graphic summaries, etc. Education programs will include artificial intelligence. They will act like intelligent teaching assistants. Specifically, programs will be more able to diagnose each student's learning problem and to provide more specific tutoring to correct it. So if three students are having difficulty with meiosis but for different reasons, the program will tutor each differently. It may emerge as the paradox of our time that the impersonal computer is able to provide our many students with a more personalized education!

Most important, the future of computers in bioeducation is up to us! Our attitude will determine if computers will go the way of programmed learning and educational television, or if they will become a viable and valuable component of the specific educational system operating in our course.

REFERENCES CITED

Crovello, T.J. 1982a. Computers in biological education. The American Biology Teacher 44: 476-483.

Crovello T.J. 1982b. CIBE systems. The American Biology Teacher 44: 374-377.

Crovello, T.J. 1983. Educational Computing Systems: The COMPEACE Guide To Evaluation Of Their Hardware, Software and People. Available for $6.50 from COMPEACE, Box 554, Notre Dame, Indiana 46556. 80 pages.

Crovello, T.J. 1984. Evaluation of educational software. The American Biology Teacher 46: 173-177.

BASIC PROGRAMMING FOR STUDENTS WITH A LIMITED MATHEMATICS BACKGROUND

By: Jerry Cummins, Proviso West High School, Hillside, IL

Teaching BASIC programming to students with a limited mathematics background requires some careful thought.

Starting the course can be a challenge. Many students with below average backgrounds in mathematics that take a course in BASIC programming may not understand the difference between programming jobs and other data processing jobs in areas such as word processing. Therefore, our first assignment is from a major "large city" newspaper. Students are required to look up employment opportunities in the computer area of the ads section of a newspaper. They are to collect at least 6 ads and tape them to a sheet of paper to be handed in. These are used for a class discussion on the different job opportunities available in the field and the education (especially mathematics) required. Salary helps students realize the differences between the variety of jobs available. This is a good time to discuss reasons for students to increase their knowledge of mathematics. This information will be used later in the course.

Regardless of the text used, first "hands-on" experiences should be with a computer in command mode. The following examples illustrate some of the appropriate explorations.

I. Explore arithmetical operations including order of operations

```
PRINT 5 + 4
9

PRINT 5 + 2 * 3
11
```

II. Explore exponential notation

```
PRINT .0000007321
7.321E-07

PRINT 1E04
10000
```

III. Explore variable and string relationships

```
PRINT "A"
A

PRINT A
0
```

IV. Explore computer rounding

```
PRINT 1/(6∧2-36)
o7108864 (ANSWER WILL VARY DEPENDING ON MACHINE)
PRINT 1/(4 ∧2-16)
DIVISION BY ZERO
```

There are many other types of explorations possible in command mode.

Line numbers is a natural topic after command mode. Students enjoy creating designs using their initials. If printers are available, this offers an opportunity to become familiar with printing procedures. The following program illustrates a design program that creates an arrow.

At this point students are ready to write letters using PRINT statements. The ads collected during the first week of class can be used as references for job applications.

A natural progression leads to programs involving LET, READ-DATA, and INPUT. I find that practically all students with a limited math background have had experience with consumer topics. Therefore, I suggest a "strand" of programs centered around "interest" problems.

I. Simple interest for one year using LET

```
5  REM SIMPLE INTEREST - ONE YEAR
10 LET P = 1864
20 LET R = .12
30 LET I = P * R
40 LET NP = P + I
50 PRINT I,NP
60 END
```

II. Simple interest for 2 years using LET

```
10 let P = 1864
20 LET R = .12
25 LET T = 2
30 LET I = P * R * T
40 LET NP = P + I
50 PRINT I,NP
60 END
```

III. Simple interest for 2 years using READ/DATA

```
10 READ P,R,T
20 DATA 1864, .12, 2
30 LET I = P * R * T
40 LET NP = P + I
50 PRINT I,NP
60 END
```

IV. Simple interest for 2 years using INPUT

```
10 INPUT P,R,T
20 LET I = P * R * T
30 LET NP = P + I
40 PRINT I,NP
50 END

RUN
?1864,.12,2
 447.36         2311.36
```

Using the INPUT statement, students can RUN the program an appropriate number of times to compound interest semiannually, quarterly, and monthly. They must

use the results from each previous calculation for data to be used in the next calculation. This leads to the following program, with documentation, to figure compound interest in a general situation.

```
5    REM EXAMPLE 2
10   REM PROGRAM AUTHOR: MARIA HERNANDEZ
15   REM PROGRAM NAME: COMPOUND INTEREST
20   PRINT "COMPOUND INTEREST PROGRAM BY MARIA HERNANDEZ"
50   PRINT
60   PRINT "ENTER PRINCIPAL,RATE AS A DECIMAL"
70   PRINT "TIME IN YEARS AND HOW OFTEN INTEREST IS COMPOUNDED"
80   PRINT "(USE 2 FOR SEMIANNUALLY, 4 FOR QUARTERLY"
90   PRINT "12 FOR MONTHLY AND 365 FOR DAILY)"
95   PRINT "TYPE A COMMA BETWEEN THE NUMBERS"
100  INPUT P,R,T,CP
110  REM  THE EXPONENT N IS CALCULATED BY MULTIPLYING THE
115  REM  TIME IN YEARS BY THE NUMBER FOR HOW INTEREST IS COMPOUNDED
120  LET N = T * CP
130  REM  PR IS THE RATE FOR EACH TIME SPAN
140  REM  FOR EXAMPLE: IF THE RATE WERE 12% AND THE INTEREST
145  REM  WERE COMPOUNDED QUARTERLY (4 TIMES A YEAR) THEN
146  REM  PR = .12/4 = .03 (3% INTEREST FOR 3 MONTHS)
150  LET PR = R / CP
155  REM  STATEMENT 160 IS THE COMPOUND INTEREST FORMULA
160  LET I = P * (((1 + PR) ^ N) - 1)
180  PRINT
190  PRINT "PRINCIPAL"; TAB( 11);"RATE";TAB( 16);"TIME";
195  PRINT  TAB( 21);"COMPOUNDED"; TAB( 31);"INTEREST"
210  PRINT P; TAB( 11);R; TAB( 16);T: TAB( 26);CP;
211  PRINT  TAB( 31);I
220  PRINT
240  PRINT "TYPE RUN TO DO ANOTHER PROBLEM"
250  END

RUN
COMPOUND INTEREST PROGRAM BY MARIA HERNANDEZ

ENTER PRINCIPAL, RATE AS A DECIMAL
TIME IN YEARS AND HOW OFTEN INTEREST IS COMPOUNDED
(USE 2 FOR SEMIANNUALLY, 4 FOR QUARTERLY
12 FOR MONTHLY AND 365 FOR DAILY)
TYPE A COMMA BETWEEN THE NUMBERS
?1864,.12,2,365

PRINCIPAL       RATE          TIME        COMPOUNDED          INTEREST
1864            .12           2           365                 505.515194

TYPE RUN TO DO ANOTHER PROBLEM
```

Problem solving needs to be approached carefully. Concepts previously covered in mathematics classes can be used for a problem basis. Computational aspects of the Pythagorean Theorem offer an arena for such problems. Following is a modification of a problem that appears in Lowell Carmony's book on Pascal programming:

"A VICIOUS, BUT HIGHLY TRAINED DOG IS AT THE SOUTHWEST CORNER OF A SQUARE GARDEN. INDIANA JONES IS IN THE CENTER OF THE GARDEN. THE DOG WON'T ATTACK JONES IN THE GARDEN, NOR WILL IT ATTACK JONES ONCE HE HAS LEFT THE GARDEN. BUT THE DOG WILL RUN ALONG THE WALL AND TRY TO BITE JONES AS HE CROSSES THE WALL. THE DOG RUNS AS FAST AS JONES. CAN JONES ESCAPE?"

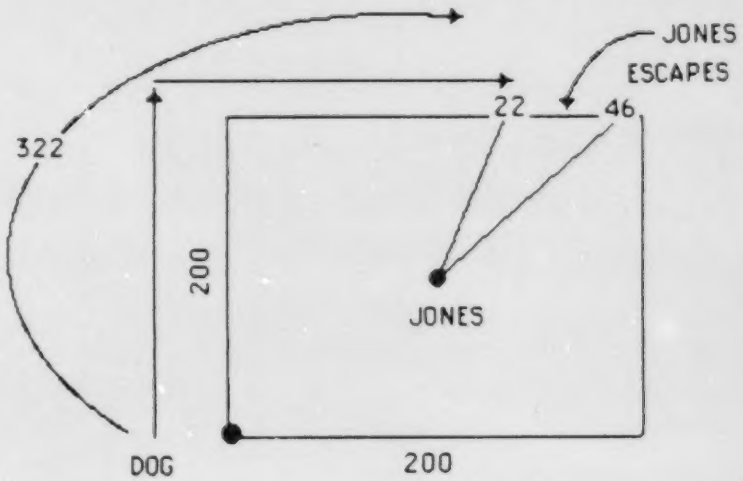
Students can hand calculate many of the results for test purposes. A program listing below will lead to the solution.

```
40   PRINT "PT";"  ";"MAN";"          ";"DOG";"  ";"RATIO"
45   PRINT
50   FOR PT = 2 TO 100 STEP 2
60   LET MAN =  SQR (100 * 100 + PT * PT)
70   LET DOG = 300 + PT
80   LET RIO = DOG / MAN
85   IF RIO < 3.14159 THEN 100
90   PRINT PT;" ";MAN;" ";DOG;" ";RIO
100  NEXT PT
110  END
```

```
RUN
PT  MAN      DOG  RATIO

22   102.391406 322 3.14479518
24   102.839681 324 3.15053486
26   103.324731 326 3.15510137
28   103.84604  328 3.15852198
30   104.403065 330 3.16082674
32   104.995238 332 3.16204817
34   105.621968 334 3.16222097
36   106.282642 336 3.1613817
38   106.976633 338 3.1595685
40   107.703296 340 3.15682075
42   108.461975 342 3.1531788
44   109.252002 344 3.14868371
46   110.072703 346 3.14337697
```

200 × 200 SQUARE USED IN THE ILLUSTRATION



BASIC programming is a good course for students with a limited mathematics background if the pace and problem selection is carefully considered. It will illustrate to students that mathematics has practical application. Hopefully, it will encourage students to take more mathematics as they realize limits in their learning capacity without it.

WORD PROCESSING: INTRODUCTION IN THE COMPUTER LAB

By: Ann Dana, Hinsdale School District #181, Hinsdale, IL

The goal of District 181's computer program for the 1984-85 school year has been to **ensure** that **all students** receive **similar instruction** in the **use** of the **computer** as a **tool.**

The microcomputer can only change education, how and what students learn, when it is used because it can do the task better than any other method. Some of the reasons for using the microcomputer is that it can store information, retrieve, sort and print data according to our own specifications and can make the task of writing much easier.

District 181 has begun a longevity study of the effects word processing will have on students' writings. Samples of students' writings were taken this fall. Periodically over the next 2 years additional samples will be collected. All of these samples are to be composed on the microcomputer. The samples will all be judged at the same time by the same team of trained personnel.

Literature on word processing appears often in magazines and professional journals. Margaret Withey in The English Journal of November 1983, stated that showing students how writers turn technology to writing something with content, write enough and organize the writing is a real use of the computer. She summarized how English teachers view the computer much like math teachers viewed the calculator, "math teachers were still arguing about whether or not students should use calculators after every student had one." Peter McWilliams says word processing on the computer offers the best method of "recording the processing of the mind." In the latest (January) Family Computing, one parent wrote that helping their son with his writing was the main reason for buying a family computer. On and on the articles go, but perhaps the most important reason is to listen to the students **ooh's** and **ah's** as they discover what a word processor program can do for them as they learn the editing process.

The District 181 K-6 students are being introduced to word processing by the same person, in similar circumstances with the same materials. This was brought about after consulting with the teachers who piloted our district computer curriculum last year. They concluded that trying to teach word processing, data processing and programming in the classroom with 1 or 2 computers was much too time-consuming and a very inefficient method of teaching these uses of the computer. It was concluded that any new hardware purchases for 1984-85 would be kept together as a traveling lab.

My position as computer consultant was changed to ensure that the lab would be my main thrust. I travel with the 7 new Apple IIe's that are added to the computers (4 to 7) that already exist in each of the seven K-6 buildings. I spend 2 weeks at each building each semester. All students receive three 30 minute sessions in the lab per week.

The Lab has proven to be challenging but exciting. The students are all receiving the same treatment and all of their teachers are receiving the same

in-service. That was one of the criteria for having the travelling lab, all teachers would attend the sessions with their students. The students are very enthusiastic. They like being able to be in the lab with everyone in their class on a computer. Computers are shared in grades K-4 and grades 5 and 6 are split so that each student has a one-on-one experience. This was deemed important as these students are also introduced to a data base.

The Lab format is similar at each school. When the students attend their first session, we find out just how the computer works and then all are introduced to a keyboarding program. Literature does state that this is an important concept as well as rumors we hear from business education teachers and listening to parents' concerns. The programs the students use are:

K-3 **The Friendly Computer**, MECC
4-6 **Type Hype**, COMPUNIQUE
4615 Clausen Ave.
Western Springs, IL 60558 ($39.95)

The great part of **Type Hype** is that you set it for 144 characters per turn for the beginner. It's just enough experience to keep them interested. There are lines or paragraphs of practice for each of the key sets being learned and teachers can input their own lessons. Dick Dlugo, the author, is a teacher at St. John of the Cross School in Western Springs. He'll give you a rate for multiple copies. See the latest issue of C.H.I.P.S. (Vol. 4, #2 MICRO-IDEAS newsletter).

The Friendly Computer does not teach the method of which finger to use on which key, but it does familiarize the younger student with the keyboard. There are 2 programs we use, The Keyboard and ZeeBug. They LOVE ZeeBug!!

All students are introduced to word processing using MAGIC SLATE by Sunburst. It's a word processing program that "**grows with you**"!! Its creative capabilities are endless. You can UNDERLINE, use **BOLDFACE**, ITALICS, OUTLINE, superscript, and subscript, all with ease. The control key, in conjunction with other keys, does such important things as DELETE, INSERT, INDENT, CENTER, REPLACE, SEARCH, and move BLOCKS of text. The MENU has eight options and each can be quickly accessed with a move of an arrow or typing the first letter of EDIT, LOAD, PRINT, NEW, SAVE, DELETE, MAKE or QUIT.

Probably the most unique feature is the ability of the program to be used in oversize type in the 20 Column Version for the Primary Student. This is the main reason why we can introduce even Kindergarten students to word processing. It is easy to read, just like a page from a Primary reading book. The regular print, just what you are seeing here, is the 40 Column Version. It is easy to switch from one to the other using Control C while the program is booting. Sunburst will release a professional 80 column version as soon as they get the printer bugs out. For $65.00, it's the easiest word processor I have taught to young students.

The manual that comes with MAGIC SLATE is extensive. It has booklets to duplicate for teaching or as tutorials. It lists writing projects for both versions (10 to 14). Each of the lessons can be used to reinforce the editing capabilities of **MAGIC SLATE.** The manual answers any question you could possibly have on how to use the program in your classroom. It's ideal to use in the lab because you only need one copy to boot in each computer. Each

student then inserts a data disk. The same data disks can be used for the 20 or 40 Column Version. As a matter of fact, you can save in 20 column and print it in 40. Printing has proven to be the only problem that is sometimes hard to overcome. At this time, the only printer we have been able to use with the 20 Column Version (it prints entirely in **graphics**) is the Epson FX with a Grappler+ interface.

Projects have been interesting and varied for each grade level. Kindergarten students normally type their names and phone numbers. They make a class list that is duplicated for each of them. It's rare but John, age 5, wrote his own version of the 3 Bears, entirely without help. With encouragement, where might he someday be? After first graders practice their names, family members, colors, addresses, we use their reading words to write sentences. They are just learning that capitals start sentences and periods or question marks are end markers. They are thrilled with their own printouts. They learn to back space to typeover.

Second and third graders are using two of the lessons in the manual, ABC order (A is for Apple) and Rhyme Time (Fish, fish in a dish!). They learn to use Control D to Delete and Control O to Underline. They write about their favorite book, underlining the title. Having each student put their name at the top of the computer screen gives them an opportunity to practice capitalizing and gives me a chance to learn their names, and also to know whose paper is whose once they're printed out.

Fourth graders use the 40 Column Version to write paragraphs; subjects are decided by their classroom teachers. The paragraphs are saved. The next session gives the opportunity to learn INSERT. The students find a noun to modify. This is first taught in a practice sentence. Type `The dog ran.' The students then move to dog, use Control E to get to INSERT mode then type a word to describe the dog. This helps them to see how words can be added. Fourth grades also use Delete, and are introduced to the other Typestyles with Control O. As with 3rd grade, they learn to underline book titles.

Fifth and Sixth grades have less time with **MAGIC SLATE** as they also learn to create a form, add data, do a search with PFS (Personal Filing System). They create paragraphs, edit with Insert, Delete, Replace and other Typestyles. They can't believe how easy it is to move around and change words or letters. It isn't enough practice but as their teachers also have learned how to use the program, they will be using it in their classrooms. Another project that the students experience is to write Friendly Letters. One class wrote to their parents telling them all about the crayfish they were studying in Science. Another wrote about Outdoor Ed and yet another about visiting the Art Institute.

Teachers at each K-6 building receive in-service in **MAGIC SLATE** at lunch time sessions. Many of the teachers also received a half day of released time in October to learn the program. Teachers are finding that they can create work sheets easily, notices to go home, news articles with added interest (using the Typestyles), assignments on disk for students to correct (on disk). Once class has begun to correspond with Pen Pals in California using **MAGIC SLATE**.

These 3 to 5 learning sessions are not enough to become proficient but each student has had a similar experience. With the **MAGIC SLATE** Tutorial Manual on

the computer table each student can quickly review what they might not remember. Like anything else we learn the more practice the better we'll be.

This has been a very worthwhile experience for the students and teachers. The **MAGIC SLATE** Manual states that word processing encourages editing. "Polished writing without drudgery is what word processing with **MAGIC SLATE** is all about." It's fun **to use** while DOING SOMETHING WORTHWHILE!

REFERENCES

McWilliams, Peter "A Layman's Guide to Word Processing", Family Computing, New York: Scholastic, Inc. December 1983 (Vol. 1 No. 4).

Pollitt, Alyce Hunt "Warming to the Wonders of the Word Processor: An English Teacher's Introduction to the Computer", The Computing Teacher, Oregon: ICCE, May 1984 (Vol. 11 No. 9).

Sullivan, Nick "Connecting Home and School", Family Computing, New York: Scholastic, Inc. January 1985 (Vol. 3 No. 1).

Withey, Margaret M. "The Computer and Writing", The English Journal, November 1983.

"Why Use a Word Processor?" MAGIC SLATE, Pleasantville, New York: Sunburst Communication, Inc., 1984.

MICROWORLDS AND MACROWORLDS

By: Andrew David, Chicago, IL

Between Two Worlds

When you look at an old-style analog watch with hour and minute hands you see a circle. You see an entire half of the solar day and where you are in that day at once. You identify your place in time by constantly referring to the larger picture. Your reference is spatial, almost intuitive. In contrast, today's digital watch asks you to think about only the slice of day at hand, divorced from the rest; the digital sensibility is to be narrow and precise, to deal with fragments, with ever more isolated views. Hands sweep across a circular face as the earth revolves within the apparent orbit of the sun; but the incremental advance of the digital display has only itself as a reference.

As we increase our reliance on digital technologies, we consign to an invisible "microworld" more and more of the problems native to our own familiar "macroworld." The advantages of moving in this direction are not trivial. But as the computer becomes a more powerful and invisible component of the things it manages, it will become more important for people to understand that there are differences between these two worlds. The means of communication between the two will need to be fluid and rich. The first thing every teacher and child should understand about these super-machines is that they can do many, many things for us, but only those things that can be expressed in a language which is mutually understood.

The Software Link

Of the computing machines currently accessible to the educational community, most do roughly the same things. They are, by themselves, essentially "faceless" carriers of untapped computer power. Each piece of software, however, transforms these machines into specialized devices by providing a selective window through which we can carry out limited manipulations. We have the beginnings of a mutual vocabulary. At the same time, the software gives them "faces" upon which we fix our attention. Gertrude, Rocky, and the Logo turtle, and even the menus at the top of the Bank Street Writer screen have all been concocted to fill our world with friendly, understandable things as we unwittingly throw thousands of binary switches each second. Looking at software in this way suggests two important questions for evaluating the quality of an individual piece:

(1) How much computer power does it access?

(2) How successfully does it incorporate a metaphorical component to engage the learner?

The two qualities go hand in hand. A program that liberates computer power without engaging the learner is incomprehensible and cannot be used. On the other hand, if it does not truly access power, no matter how embellished the screen, it can reduce a potentially valuable tool to glorified television.

A Context for Logo

Logo does well on both counts. It accesses a wide range of the computer's capabilities (probably a good deal more than most users discover during regular use) while employing the metaphor as its main delivery vehicle. The turtle is, of course, the most well-known and highly developed Logo metaphor.

It is both an obedient animal and pixel-configuration. It responds to familiar words and behaves in predictable, "natural" ways that are, in fact, complex subroutines created at a high cost to memory. It allows us to navigate the monitor screen as freely as our own living rooms, without having to think about X and Y coordinates, though we can if we so choose. In a sense, the turtle is at home in both worlds. It provides a link, or intersection, between computer and person just as any metaphor provides a bridge between two different levels of meaning. This intersection defines a domain of mutual understanding; where there is mutual understanding there is communication and where there is communication there is action. We can do something.

What is appealing about Logo is that the intersection is fluid. It grows every time a procedure is defined; it molds itself to fit the individual. On the inside, it is intended to be flexible as well, so that it can mold itself to fit the machine, though not so readily. It is like a pivot point, the narrow part of an hourglass. For this reason, we may choose Logo as a means with which to explore the difference between machine experience and human experience, in order to discover how much a computer can do that is meaningful to us. The challenge of integrating Logo into the curriculum is a special application of this process.

How Much Experience Is Computable?

This question, of course, can never be answered. But we can keep pushing back the limits of what we perceive to be "computable." First, we need to define the term for ourselves.

When we learn about FORWARD and RIGHT, we discover two attributes that can be represented by the computer: position and heading in two-dimensional space. We have also discovered a line-tracing method. Together, these unlocked powers enable us to solve certain problems. Squares, spirals, houses, and primitive trees can all be drawn; in fact, any continuous-line drawing can be crafted. Learning about PENUP and PENDOWN opens up a new area of problems that can be solved: the representation of drawings with discontinuous lines as well. In fact, every new primitive we learn unlocks a new capability of the computer, thus opening up new areas of problems that can be solved. This suggests one definition of "computable experience":

> Any experience is computable if it can be represented, depicted, or managed with computer capabilities to which the user has access.

Which, in turn, suggests a methodology for selecting meaningful activities:

(1) Decide what you want to do. What themes or subject areas do you want to explore?

(2) Freely explore the Logo environment in order to discover new capabilities. What kinds of things does it allow you to make the computer do?

(3) Alternate between steps (1) and (2). With each discovery, the two processes will modify each other. In what ways can the new capabilities be applied to your chosen subjects?

This last question presents a significant challenge. Successful solutions require creativity and insight. And time, something few of us have in excess. Remember, the MIT research team had nearly 15 years to develop the first generation of Logo implementations although, in the earlier stages, they had nothing like today's personal computers to work with. Someone wrote recently that Logo is not so much a tool as a machine shop. You have to build your vice and hammer before you can build your bookshelf. And making a vice is a much more difficult problem. But at least you don't have to mine the ore.

The fact is, there are people busy building vices - in many fields other than computer science - and if their work can be disseminated, either through networking or marketing, it is possible that a much more productive period of individualized programming will follow. Dissemination through marketing is the more common process, though a characteristic of the Logo community is its tendency towards free networking. Logo is helping to broaden the base of "computer aware" individuals; it joins other powerful software tools in appealing to a diverse population, creating the possibility for an exponential increase in the kinds of experience perceived as computable.

The Evolving Workstation

Another factor affecting perceptions of what is computable will be significant changes in computer-related technologies. My crystal ball does not reveal a "computer" per se; rather, it depicts the computer disappearing into the immediate environment it manages - the "computer workstation." Already, we are seeing the arrival of speech synthesizers, musical keyboards, and so-called intelligent videodiscs, as well as a preliminary generation of personal robots. It seems likely that the computer will evolve to become:

- More powerful
- More sensory
- More modular
- More standardized
- More invisible

leaving aside, for now, the central issue of intelligence.

How can we prepare for such changes? One step at a time. Imagine getting into a car and saying

```
TO HOLIDAY
    SETHEAT :LOW
    SPRAYWASH WIPER 3
    GETMAPS [CHICAGO ILLINOIS WISCONSIN]
    DISPLAY [CHICAGO]

    etc.
```

Of course, by the time we all have these "intelligent" cars, we won't be speaking in such broken English, but it would be nice if procedural capabilities were still around and if we were well-versed in using them.

At any rate, this is an example of anticipating possible futures with more limited techniques and resources currently accessible to us. In general, one possible way to predict and prepare for the near future is:

(1) Keep aware of current state-of-the-art technologies including those still at the research and development level. (We are now working with hand-me-downs from yesterday's state-of-the-art. Today's personal computer is tomorrow's wristwatch; today's supercomputer is tomorrow's desktop.)

(2) Look at what you now work with. Find the overlap between that and what you can expect in the next generation of resources. Minimize your focus on those things which will become quickly obsolete. (Five years from now, we won't be be pressing "<Control> G" to break into a program, but we will always need an "abort function".)

(3) Find ways of utilizing your current resources to create, in microcosm, what you can anticipate with more sophisticated resources.

As an extreme example of (3), I use a carousel slide projector and a library of slides in anticipation of the day anyone is able to call up images from a videodisc. Disc images will appear on monitors and be accessible for image processing. In my home-made version, the slides are projected onto graph paper and traced as pixels, thus digitizing the image so it can readily be entered into a computer. (In Logo, for instance, it can be defined as a sprite in TI, Atari, Commodore, or Sprite Logo implementations.) There are image digitizers available now, costing not much more than a carousel, but the carousels are already a standard part of many schools, including the disadvantaged inner city public schools I have worked with. The slide projector's potential falls short of the videodisc's in significant ways; but images on disc are not pervasive in the culture - whereas one can obtain a slide image of almost anything. And images are an inroad to content. Too often, the love affair with technology assigns secondary importance to content.

From Macroworld to Microworld (And Back Again!)

If one of the teacher's major challenges is to recognize problems which are both relevant and "computable," the investigation of future technological advances may shed light on one fundamental consideration:

> Be sure to distinguish between limitations of knowledge and limitations of technology.

Many images, for instance, may appear beyond the scope of Logo for accurate duplication. Imagine a lifelike rendition of cylinders, cones, and spheres, stacked, glistening in the sun, painted with swirls, in front of a brilliant blue sky with puffy white clouds. It can't be done. But take away, say, the clouds in the background (may be done easily when enough memory becomes available for high-powered "fractal" programs), the brilliant multi-colored swirls (we need only await a more versatile graphics chip to get the full range of colors), the sharpness (high resolution TV is upon us), and the base colors from each area (some versions of Logo already have a FILL procedure). What remains? A skeleton, a two-dimensional continuous line drawing! A typical day's work for the turtle. Still difficult, because the foreshortened spheres read as complex ellipses, but this is a limitation of knowledge, and isn't it exactly the educator's job to help overcome these? We took a discouragingly complex problem and saw through the technological limitations

to the "computable" skeleton where lack of knowledge is the only obstacle. Ten years from now we can add on the peeled layers.

Eventually, the concept of "computability" may disappear. We write and speak all the time without consciously considering whether our statements may be "symbolically represented." What happens to us when we peer through the geometric objects to uncover their "computable" skeleton? How is our vision sharpened? Will it affect the way we view two renditions of mother and child side by side, one an early Picasso, the other a Cassatt? Will it add to or take away from our appreciation? How will our voyages through microworlds affect the way we view the macroworld?

PROFESSIONAL TRAINING IN CBE: THE VIEW OF ONE UNIVERSITY

By: J. Richard Dennis, University of Illinois, Champaign, IL

As demand for conventional teaching personnel has diminished, colleges of education have sought alternate audiences. The rapid expansion of computer-based teacher/training in all sections of society has presented one area where reconfiguring of education courses and programs has been particularly productive. An instance of one such evolution is presented here to outline career opportunities and related training.

A Brief History

The College of Education, University of Illinois (UIUC) initiated opportunities to learn about computer applications to teaching in 1970. It is no accident that this institution should have spawned such early efforts. By 1970 the Urbana-Champaign campus already had logged a long history of computing technology research growing out of the various ILLIAC computer projects. Also, by 1970, there had been nearly a decade of exploration, development, and research on early computer-based teaching centered on the now famous PLATO project.

The intervening years have seen a steady evolution in numbers of courses, breadth of interest of related full time and part time faculty, and of the intensity and level of sophistication to which students are prepared.

Careers and Training

The UIUC currently offers training to pursue the following kinds of computer related education/training careers:

1. Pre-service preparation of teachers of conventional subjects, with opportunities to specialize as a computer-using teacher of those subjects.

2. Pre-service preparation (to recognized certification) of teachers of computer science for pre-college levels.

3. Graduate, in-service preparation of teachers of conventional subjects to become computer-using teachers.

4. Graduate degree preparation of teachers of computer science (master's level).

5. Graduate preparation of college/university faculty in computer-based education (Ph.D. and Ed.D degrees).

6. Graduate preparation of materials/training designers for industrial/professional training or educational publishing careers.

7. Graduate preparation of computer resources managers/ coordinators for schooling systems and institutions.

8. Graduate preparation of consulting or facilitating personnel.

These career profiles have similarities and differences. The threads running through all of them are education/training theory and practice, computer technical expertise, and generic preparation in using interactive electronic devices as teaching media. The differences are determined by background and experience of the individual, and career goals.

A more intimate perspective of these related careers can be found by reviewing specific course titles involved in the various degree programs. We have omitted the entire catalog of technical computer science courses, and the more remotely connected (but still frequently relevant) listing of educational research/evaluation courses. At this writing, the computer-based education courses menu looks like this at UIUC:

Computer-Assisted Instruction
Computer Use in Education
Application of Computers to the Teaching of Humanities
Application of Computers to the Teaching of Science
Application of Computers to the Teaching of Social Studies
Application of Computers to the Teaching of Mathematics
Computer-Based Foreign Language Teaching
Computer-Assisted Instruction in Elementary Education
Instructional Design
Instructional Design for Computer-Assisted Instruction
Interactive Systems for Instruction
Interactive Systems Design
Computerized Adaptive Testing
Theory & Design of Instructional Simulations
Cognitive Aspects of Interactive Instruction with Computers
Introduction to Artificial Intelligence
Applications of Artificial Intelligence
Computer Inference and Knowledge Acquisition
Knowledge-Based Programming
Pattern Recognition & Machine Learning
Fundamentals of Curriculum Development
Continuing Education Program Development
Adult Learning & Development
Introduction to Management Information Systems
Information Organization for Management Information Systems
Management Information Syster Development

Further information about these degree opportunities may be obtained by writing: Coordinator, Studies in Computer-based Education, 395 Education Building, 1310 South Sixth Street, Champaign, IL 61820.

KINDER KEYS, A KEYBOARDING PROGRAM

By: Mary Ann Engebretson, Stillwater Public Schools, Stillwater, MN

KINDER KEYS is an enjoyable, effective method of introducing keyboarding. The program is an active learning process that associates letter names and numbers with interesting graphics and music. At the same time, correct character placement on the keyboard and correct hand usage are introduced. The program assumes no prior computer use and can be used by anyone ready to learn letter names and numbers.

There are three major ways in which elementary school children will be taught to keyboard - in labs where all children are learning to keyboard at the same time, in small groups while other activities are going on in the classroom or media centers, and individually in a classroom or at home.

The program is intended for students in kindergarten through second grade, but can be used by any student ready to learn letter names.

The program is an active learning process that provides students with the opportunity to learn letter names and numbers associated with interesting graphics and music. At the same time, students learn correct character placement on the keyboard and correct hand usage.

Characters are introduced in the logical sequence of placement on the keyboard. Introduction follows character by character in a left to right, top to bottom sequence, thereby correlating with any reading readiness program in use. This sequence also permits numbers to be introduced first, then letters, and finally special keys.

Lessons are presented in groups for the left hand and groups for the right hand on each row of the keyboard. This prepares the students for future instruction in touch typing.

The program assumes no prior computer use by the students. The menu displays the lessons.

A large graphic character set was designed for use in this package. This character set simulates the primary print used by several reading readiness programs. It also permits the use of upper and lower case characters. Letters and words are introduced in lower case and the corresponding upper case letters are also displayed.

The students are required to press each key 3 times. After the first time, they are reinforced with a graphic and sound for knowledge of results. After the second and third time, they are reinforced with a short, pleasant "beep" for knowledge of results.

The second half of each lesson consists of an incidental learning activity. In the number lessons, students must count the number of houses and key the correct number. In the letter lessons, a puzzle consisting of portions of the graphic reinforcement for the first half of the lesson is presented.

At the end of the lesson, the students have the option to Go Back to review the lesson, Go On to the menu to use another lesson, or Stop. To further enhance the program, several activity sheets have been developed for use with this package.

Computers in Our Neighborhood Coloring and Activity Book is a coloring and activity book which is intended to be an enjoyable, informative introduction to computer careers. The coloring book is compiled as a field trip throughout a hypothetical neighborhood. The careers begin with ticket agent from whom the class gets tickets. The teacher's information packet provides background on computer use by all careers in the coloring book. The field trip progresses through the neighborhood with activity and coloring pages provided along the way. All careers incorporated in this learning activity are real and lend themselves to lively and interesting discussions. This is a resource that is completely assembled. The teacher may reproduce it as it is to be used in class. No additional materials are necessary - not even a computer.

LASAGNA CODE...AN ALTERNATIVE TO SPAGHETTI

By: Annie Gjertsen, Buffalo Grove High School, Buffalo Grove, IL

During a three week introductory computer course offered last summer at Buffalo Grove High School, a co-worker and I decided to take a team approach to the course. Frustrated with the wide range of abilities of students in our introductory classes (everything from learning disabled to calculus students), we decided to cover our district curriculum for literacy in much the same way as we had in the past. But, for the programming aspect of the course we decided to try a structured approach.

Our introductory computer course is a synthesis of computer literacy topics. With 50 students enrolled, we split the group in two--working most days as half lab, half classroom activities. Shelly Cashman text, chapters 1, 2, 3, 5, 6, and 13 acted as reading and reference materials for the students. MECC software served as an introduction to our hardware and lab setup. Ethics were covered through the viewing of the Phil Donahue show on the Milwaukee users group that broke into businesses through computers. Bank Street Writer was utilized as a word processor for magazine summaries. Data entry and data inquiry activities used accompany our textbook. The remainder of the curriculum covered flowcharting, trace tables and structured basic programming.

A structured program is like an outline one does prior to writing a paper: Roman Numerals denote headings, each heading broken down into subheadings, and these further subdivided. The goal is simply to break a task into its subtasks. Any problem can be solved by repeatedly breaking it into subproblems, until every subproblem can be solved.

You do not need a structured language like Pascal to teach good structured concepts. Once we maintain control of the structured process, any language may serve as the vehicle for coding. This is the approach we stress. Structured programs tend to be:

1. more readable and understandable
2. easily modified and maintained
3. more dependable
4. easily debugged
5. easier to document
6. written with clear code

We have found that once the structured process and its concepts are understood, they carry over into higher level programming courses and into other academic areas as well is the pay off for the time and effort in learning a new approach.

SOFTWARE EVALUATION IS A PERSONAL AFFAIR

By: Mrs. Bobby Goodson, Educational Consultant, Sunnyvale, CA

Software is evaluated by a variety of people, in different ways for many purposes. It is important that we recognize this and use all of the tools at our disposal as we select the software we are going to use within the educational process. It is also important that we recognize the different types of evaluations and accept the information they provide with understanding.

The first step in locating software often involves the use of reviews. These arrive in the mail from developers and are found in many of the popular computer and educational magazines. Reviews are even found in some distributors' catalogs. While most reviews are found in print, it is not at all unusual to get the review of a piece of software verbally at a conference, an inservice class or from a friend. A review really just describes the software, its strengths, its unusual features, and occasionally its shortcomings. The first reviews we see, often the ones in popular magazines and distributors' catalogs, are little more than a reflection of the developers' descriptive material that accompanies the software's release. The reviews that appear a little later in some of the professional journals give us a bit more information and may well provide a good, objective description of the product. These reviews serve a real purpose and help to identify those pieces of software we want to investigate further, but there is seldom enough information to make an informed purchase. We may very well, however, decide that a piece of software warrants a closer look and arrange to get it for evaluation on the basis of these reviews - or we may wait until evaluations, or what is sometimes called critical reviews, appear in the professional journals.

A critical review, or evaluation, should report on the use of the material with students, in the classroom or other educational settings. A really helpful evaluation is done by someone who understands the learning process, the curriculum and the student. The description of the program will relate its operation and its content to instruction. Since these evaluations are based on field tests or lengthy classroom studies, they often do not appear until the software has been advertised for some time.

While reviews tend to give us a description of the program and its operation, an evaluation will provide more details. After reading an evaluation we should feel sure that the program operates easily, without error--that technical problems have been eliminated. If the evaluation is sufficiently detailed we should also feel confident that it is instructionally sound and factually correct. The evaluation should provide information about the adequacy of the instructions and the quality of the support material that accompanies the package.

Well done evaluations provide us with much of the information we need to make our software selections. There is, however, more information that is helpful and is not always found in these journal articles. There are now several commercial software evaluation organizations that gather, evaluate and report on a great number of packages at regular intervals. These evaluations have the advantage of a constant format so that the information is readily available. They also often include detailed information about hardware

requirements, multiple machine availability, price, etc. They will identify appropriate grade levels, instructional mode, and suggested uses. The consistency of an individual organizations reviews makes them an easy form of reference.

With positive evaluations in hand we can certainly order the software and look forward to a new classroom experience - or can we? There is one more level of evaluation that is needed for the successful integration of software into the curriculum. This is the personal evaluation done by an individual teacher. It is a time consuming, demanding process that can insure a successful lesson. No two teachers plan their lessons or present their material in exactly the same way. We have always adapted the material we use to our own style, enriched it with our own experiences. The computer is no different in this way than any other teaching tool - except it may be a little harder to learn to use and certainly not as easy to adapt. We must then learn to be more critical about the software we chose to use.

The first steps in a personal evaluation is to assess the material incorporated within the program. Is it really suitable for the students in your class? Does the method of presentation match your teaching style or will it cause some confusion in the minds of your students? Do your students have all of the necessary skills to use the program successfully? Does the vocabulary match that which you use in your classroom? Can it bring about results that will enhance the lesson you have planned? These are not easy questions to answer and will require more than one quick look at the first few frames of the program. The time spent to look at a program in depth will, however, go a long way toward making the computer an important, successful and exciting part of your classroom.

These personal evaluations can also be considered from another perspective. There are those qualities that you want in a program to insure its success within a given lesson. This is evaluation from the teacher's perspective. In this evaluation you pay particular attention to the content, not just the accuracy but the degree to which it matches the lesson you have planned. Does the form of a simulation fit the picture of history portrayed in the text and the films you have used? Or does the vocabulary and type of sentence used in a language drill fit with the material you have already assigned in the text or workbook? Be aware of the sequence in which the material is presented. Does the pattern of math problems match that found in the text you are using, or are some of the less able students going to be confused by the appearance of a new idea before they are ready?

The style of presentation of a program often determines the way in which it will be best used in the classroom. Does this style match yours? If the program can best be used by a small group of students, discussing the decisions involved as they progress through the lesson and your classroom does not lend itself to small group activity or your computers cannot be arranged to accommodate a group of people, this software may not work well for you. There is also the question of time. If you can only allow groups of students 15-20 minutes at the computer, a simulation that takes at least a half an hour to work through will be a source of great frustration to you and your students. These are all questions that need to be answered individually with your own situation in mind. No printed or professional evaluation can really answer these questions.

The second part of this type of evaluation is the look you need to take at the software from your student's perspective. We have already mentioned the problem of appropriate vocabulary in the lesson, but it is also important to be aware of the vocabulary and reading level in the directions provided with the program. Instructions that are unclear or incomprehensible to your students will often create a problem that can disrupt the classroom and defeat the entire lesson.

The format and presentation style will often affect the way students react to a program more than the material itself. Try to picture their approach to a problem and their attitude as you look at a program and try to anticipate how they will react. If you are unsure of their attitude but the program is sound and a worthwhile addition to the lesson, make sure that you set the stage for the computer activity and make it a definite part of the total learning process. This will help concentrate the student's attention and set the stage for the positive results you desire.

Careful evaluation is the key to the successful use of computers within the normal curriculum. It is a time and energy consuming task. There is now a great deal of information available to help us in the selection process. However, we must learn where to find the information we need, make use of all that is available and devote our efforts to that final process of personal evaluation to make sure the program is right for our purposes and fits the needs of our students.

COMPUTER EXPEDITION 1984: WHAT TO DO WHEN THE NOVELTY IS GONE

By: Michael Gordon, Director, CADRE, Glen Ellyn, IL

Introduction

Most schools today believe that some form of computer literacy must be taught. The problem which faces teachers and curriculum planners is that no widely accepted definition exists as to what that means. It may be terminology; it may be programming. It is clear that there is no widespread agreement. There is some feeling that what is taught in the schools may sometimes even be harmful. Edger Dykstra: "It is practically impossible to teach good programming to students that have had a prior exposure to Basic: as potential programmers they are mentally mutilated beyond hope of regeneration."

Inservice for teachers frequently also focused on programming, most often in Basic. Almost as often the same teachers were expected to return to their classrooms and translate those programming lessons into classroom lessons for their students. It is probably true that in more cases than not, teachers returned to their class with the clear message that this technology was "For Experts Only."

Microcomputers were introduced in the schools by hobbyists who had to create any program they used themselves or by exchange with other hobbyist/teachers. Computers in the schools appeared to many (and still to some) to be only another of the fads which have come and gone on the educational scene.

The development of the capabilities of microcomputers did lead though to their thorough integration into the rest of business and society. As more and more students were found to have computers in their homes, schools began to react in a variety of ways. Some rushed out and acquired a lab of computers and then wondered what they were going to do with them. Some did plan, and then acquired and used theirs for a variety of purposes. Some are still waiting. Most schools have recognized that there is a mandate for them to deal with the problem of their role in the computer education of their students and hence they must deal with the integration of computer education into their curricula.

Context

On September 24-26, 1984 personnel concerned about the future of education gathered at the Lake Geneva campus of George Williams College. The conference was sponsored by Cadre, the Computer Technology Services Consortium of DuPage and Kane counties. The conference addressed itself to the question **"How can we cooperatively catalyze the development of computer education in the schools?"** Cadre invited Lens International to facilitate the seminar using a ThinkTank strategic planning process. Using this approach, a consensus was built which eventually lead to an action plan.

The Process

The Conference opened with position papers by persons representing elementary/secondary education, industry, and higher education.

Marguerite Cox presented On the Way to the Future. In it she reviewed the opinions represented in current literature concerning the future of computer education, reported on the computer program of Glenbard District 87 which she represents, and presented the prediction that "computing is the discipline that affects all other disciplines. It will become as necessary to literacy as books and writing implements."

Larry Blasch of IBM Corporation in his presentation of Instructional Computing, Its Direction and Delivery, focused on the challenges to education: to identify the best ways to present material to students, using computers when they are the best way; seeing that all students get access to the same educational information, and maintaining a manageable system---all at a manageable cost.

A.I. Weinzweig from the University of Illinois Chicago speaking on Where Are We Going and Where Should We Go with Microcomputers in Schools, proposed that the computers major role in education will be as an aid to learning, and teachers will need to refocus on the things that the computer can help them do better than they are currently being done, but not to use it where other means or technologies are more appropriate. (Using Logo to have a young child draw pictures with circles and squares can more effectively be done with blocks than on a computer screen.)

With the positions presented serving as stimulators the participants were drawn into the Lens process: **A Practical Vision** is created; the **Contradictions** which block the achievement of this vision are identified; **Strategic Approaches** are created to attack the contradictions; an **Implementation** schedule is created.

The Vision:
Working as a group ideas were elicited which were to be molded into a collective vision. Involving universities with schools, education at home and its relation to that received in the schools, networking among educational institutions, universal access of students to the universe of information, personalized education, new responsibilities for teachers, children treated as thinking human beings, individuals as life long learners, the computer as a tool, teachers being able to access sophisticated educational technologies were facets of the vision.

These and many more were grouped into arenas of concern: Systems Articulation, Access Policies, Instructional Processes, Curriculum Values, and Technology Tools. The arenas were plotted onto a chart which served as the focal point for the next level.

Underlying Contradictions:
Each participant was challenged to detail what they saw as standing in the way of their vision. None were so optimistic as to believe that their vision would come to be unhindered. Blocks such as an antitechnology mindset, the rapidity of technological change, inadequate facilities, scheduling complications, conflicting teacher roles, and societal pressures for immediate

results, all became part of the raw data. Nine groups of contradictions emerged in the analysis: Inadequate understanding of the long range planning process, an undefined mission for the school, unclear priorities for the allocation of resources, and undeveloped superstructure for communication, inadequate preparation and development toward teacher competency, external mandates for allocation of time, no consensus on the rewards needed to effect change, community ignorance of the benefits of technology, and limited abilities to assimilate change.

Strategic Approaches:
Having identified specific conditions which stood as blockages to the vision, subgroups were charged with proposing a strategy to deal with specific blocks. The building of consensus and participative planning, systematic progress monitoring, community input, industry linkages, user groups, resource allocation mechanisms, recognition programs, experimentation with the school year and a multitude of other strategies were proposed to deal with the blockages to the vision. As the analysis continued these approaches were seen to reside in five clusters: A Broad Based Planning Process, a Multi-faceted Communication Network, Approaches to Optimize Human Resource Development, Incentive Programs, and Comprehensive Policy Changes.

Implementation:
Each participant, aided and guided by others in the group, created a one year plan to be implemented in quarters. Each task was chosen to tie into one of the strategic approaches identified. The plan which each devised was not designed to be all inclusive; it was designed to be achievable.

Implications

One of the most surprising things to participants in the conference was that very little was a surprise. It would seem that we educators have very similar hopes for the future, and we perceive the same blocks to the achievement of that vision. What proved to be enlightening, though, was that by focusing on those blocks, specific strategies could be created --- and that led to action. The paralysis that can overtake a system does not have to occur when we orient our action planning to small, concrete, manageable steps.

Cadre surveyed the participating school districts and found that without exception, those reporting to us not only achieved their quarterly action plans, but in most cases created and achieved additional activities to help effect their strategy.

There were schools in an introductory phase who used the initial funding approved by their administrations to install their computer systems; their students were encouraged to educationally play.

There were schools in an acceptance phase were the dominant activity revolved around strategies to get teachers to say "yes" to computer education. Mobile laboratories, training teacher along with their students encouraged teachers to work with the computers.

There were schools whose teachers began to see the use of the computer as an educational tool and who looked for creative ways to use it with which to teach their subjects.

The message is clear: when there is a plan and it involves, as these did, a broad base within the school system, success is achievable. The knowledge which will move us toward that common vision exists within our systems. The answer to the question of what to do when the novelty is gone turns out to be simple. It is to prepare a plan, and do it.

EFFECTS OF MICROCOMPUTER TECHNOLOGY ON YOUNG HANDICAPPED CHILDREN

By: Patricia L. Hutinger, Western Illinois University, Macomb, IL

PROJECT ACTT

Project ACTT: Activating Children Through Technology is a Handicapped Children's Early Education Program (HCEEP) model demonstration project funded by the U.S. Office of Education, housed in the College of Education at Western Illinois University. The major goal of the project is to develop, implement, and demonstrate an innovative microcomputer curriculum model using affordable and practical microcomputer hardware and software. The primary target population is comprised of children from birth to 6 years of age who demonstrate moderate to severe structural and functional handicapping conditions which prevent them from interacting with their environment. The families of these children are also major targets for intervention and are partners in Project ACTT.

ACCT is a rural project, developed and demonstrated in a rural area of West Central Illinois. It incorporates the direct efforts of six agencies: Western Illinois University, McDonough County Rehabilitation Center, Bushnell-Prairie City Community Unit District #170, Colchester Community Unit District #180, Hancock Central Unit District #338, and West Central Illinois Special Education Cooperative. The microcomputer curriculum intervention developed by the project is being used in conjunction with the target population of children served in the McDonough County Rehabilitation Center 0-3 program, the 3 to 6 public school programs in the two school districts, and the Severe and Profound Handicapped classroom housed in the University school.

THE ACTT CURRICULUM

The ACTT curriculum has been developed according to three priorities: first, curriculum to foster the child's expectations of control of the environment; second, curriculum to provide communication; and third, curriculum to develop selected preschool skills for children from birth to six years of age. Curricular objectives have been developed to give young handicapped children who have limited interaction with their surroundings, opportunities to control or manipulate their environment. Curricular objectives for the 0-3 population have been designed to be used with the Macomb 0-3 Rural Project's Core Curriculum[1], a functional curriculum widely available throughout the country. Goals and objectives reflect a functional approach, and include fine motor, gross motor, social, cognition, language (including an emphasis on pragmatics), and self help skills. The curriculum has developed by an exemplary nationally recognized infant model project which has served hundreds of handicapped infants and their families since 1975. Microcomputer curricular activities are used with various adaptations as deemed appropriate.

In order to foster cause and effect concepts with 0-3 children, switch activated toys have been used. These toys are being interfaced with the computer in order to monitor responses, change switch capabilities, modify reinforcers, and help the child become accustomed to using a switch, a skill that leads to more sophisticated computer use at a later date. Several software programs have been developed to teach children to use switches to control the visual or sound features in a program. In addition, some commercial software has been developed to teach children to use switches to

control the visual or sound features in a program. In addition, some commercial software has been customized by adding pre-boot disks for single switch input.

Preschool curricular objectives have been developed by the ACCT staff[2]. A progression of developmental behaviors associated with acquisition of LOGO skills is currently being tested[3]. Other objectives, related to a wide variety of preschool goals as suggested by Neisworth and colleagues[4] in the HI-COMP curriculum, are related to cognition, problem solving, gross motor skills, fine motor skills, language development, social skills, and self help skills. Customized applications include the use of switches, keyboards, voice synthesizers, robots, and other peripherals. Specific adaptations that must be made with commercial software are part of the workscope. Review of appropriate commercially available software is an ongoing activity.

The 3-6 population has used LOGO extensively since it is intended to help children acquire skills that are important to ACTT goals. Adaptations to allow for alternate input have been developed, but all children do not need to use such input. A detailed LOGO curriculum is being developed.[5] The effects of the ACTT curriculum on the children are being examined carefully. Videotapes of the computer intervention sessions are taken for later analysis. The project is implementing single subject design studies in order to determine the various factors which hinder or help each child's progress. Questions related to child preferences for varying software and firmware characteristics, as well as optimal time and content of computer sessions are being studied.

In addition, an observational instrument, the BIT (Behavior Interaction Tool) was developed for use with individual children or groups of children. A program to record the behaviors and analyze them as been coded for the Epson HX20. The BIT is currently being field tested. Results are expected to provide important information about what young handicapped children actually do when they are involved in a computer experience. The BIT allows collection of data regarding individual child behavior, child-peer behavior, child-adult behavior, and adult behavior.

PHYSICAL EQUIPMENT

The computers selected for the ACTT environment were Apple II+'s and Apple IIe's. The decision to use them was based on the variety of available peripherals and software. Since some of the applications involve internal machine modifications it was important to purchase equipment that could be easily interfaced with various peripherals for specific purposes. A series of input-output (I/O) boxes have been constructed to make use of the potential of the gameport for alternate inputs. These I/O boxes can be used to accept single or multiple switch input in place of game paddle or single key control.

CASE STUDIES

David, a five-year-old child with cerebral palsy is confined to a wheelchair. While he has head control, good speech, and a cheerful sense of humor, he does not have functional hand control. His vision is restricted in ways his parents, teachers, and other professionals have not yet determined, but David can control a Topo robot with Instant LOGO and a Koala Pad and move Topo around the room in ways he plans ahead of time. He knows that he is in

control - that he can stop and start the robot whenever he wants to. With the use of an adaptive keyboard, David can make stories with Story Machine and he can play Dragon's Keep if someone reads the test to him. Using special software, he can touch the keyboard and make sound, either music or synthesized speech. He can work through the First Words program by himself. He is learning, with his mother and often with his father, to work through a variety of computer activities to help him learn to use the computer as a tool and to function in the kindergarten placement.

Four-year-old twins John and Mark, both have cerebral palsy. John is visually impaired and Mark is blind. Both twins function, physically, at a developmental level below a year, but they are socially aware and respond happily to other people. They are using switches to make things happen on the computer - sounds, music, and animation. Communication programs using speech synthesis have been initiated. The twins' mother works with an ACTT staff member weekly and their father comes to sessions when he is able. Both parents feel that the computer's use as a tool is already helpful to their children and that it will be even more useful as they grow older.

CONCLUSIONS

Young handicapped children can control their environment through computer use. Early intervention, using the computer as a tool to help the child gain autonomy, develop problem solving skills, and acquire a sense of cause and effect are part of the curriculum being developed by Project ACTT. Materials presently in the production stage will be widely disseminated to help others accomplish similar work. Data collected will provide important information about the effects of technology on the young handicapped child and his or her family. Careful application of the findings may be beneficial to a whole generation of Information Age children.

ACKNOWLEDGMENTS

This work was developed with funds provided by a grant from the Office of Special Education Programs, Department of Education. The content of this paper does not reflect the position or policy of the U.S. Department of Education, and no official endorsement of this paper should be inferred.

REFERENCES

1. Hutinger, P., Marshal, S., and McCartan, K. Macomb 0-3 Core Curriculum. Macomb: Outreach Macomb 0-3 Project, 1983.

2. Harshbarger, K., Perry, L., Robinson, L., and Sutton, A. ACTT Curriculum, in press.

3. Harshbarger, K. Developmental Stages of LOGO, in press.

4. Neisworth, J., Willoughby-Herb, S., Bagnato, S., Cartwright, C., and Laub, K. Individualized Education for Preschool Children. Germantown: Aspen, 1980.

5. Perry, L. Getting Started with LOGO, in press.

THE ROLE OF THE MICROCOMPUTER AT ILLINOIS STATE UNIVERSITY

By: Jeannie Ireland, Illinois State University, Normal, IL

Illinois State University in Normal was established as a teacher education institution. That function continues as an institutional goal, however, the goals of the university have been considerably expanded. Today, ISU houses Colleges of Applied Science and Technology, Arts and Sciences, Business, Education, and Fine Arts. Administrative personnel are encouraging use of microcomputers in all colleges and supporting such use with funding for equipment. At present, there are microcomputers in every college and in most, if not all, departments.

The information presented here is based on a survey of the 27 departments on the Illinois State University campus. Of these, 23 departments returned the survey. These data, then, represent 23 departments housing 638 faculty members.

COMPUTER EQUIPMENT

Although there is a variety of microcomputers available on campus, the IBM personal computer and the various Apple computers are most frequently used by departments. Of the 82 IBM's on campus, however, approximately 70 of them are housed in the College of Business. In this college, there is a microcomputer laboratory for student and faculty use. The Department of Music in the College of Fine Arts has an Apple laboratory and the College of Education has a laboratory housing Radio Shack microcomputers. Both these labs are available for student and faculty uses. The following table shows the number of microcomputer units in the 23 departments represented on the ISU campus.

82 IBM personal computers
79 Apple II, II+, IIes
18 TRS-80, Model III Radio Shack
16 Columbias
7 Compaqs
4 DEC's
2 Altos
2 Hewlett Packards
1 Tandy 1000, Vector, Televideo portable, Zenith, TEK, Osborne, and Ohio Scientific

Printers are important components of microcomputer systems and there are many on campus. While 59 dot matrix printers are housed at ISU, 36 letter quality printers are available. In view of the uses of the microcomputer on the campus, this is not surprising.

AVAILABILITY OF MICROCOMPUTERS TO PERSONNEL

Every department represented has microcomputer systems available in offices for administrative tasks. Advising has become simplified across campus by the installation of microcomputers which are hooked into the mainframe. Word processing is the most frequently reported use of the microcomputer with every department reporting that as the major function of the computer for them. Departmental computers are used frequently for data management in 34.7% (8) of the departments and occasionally in an additional 34.7% (8) departments. Only

13.7% (88) of the faculty use the computer for such administrative tasks. However, 8.6% (55) faculty members do at least occasionally maintain grades using the systems.

Twenty-one of the reporting departments indicated there were microcomputers faculty members could use. However, most of these are housed in departmental offices and are shared with secretarial and administrative personnel. About half--45% (10) of the departments indicate that computer equipment can be checked out by faculty for use at home during holidays and weekends. Begun as a protective measure (to cut down on thefts of computer systems), some departments now encourage the practice so faculty will become increasingly computer literate.

Students have access to at least some computer equipment in 80.9% (17) of the departments. In some areas, students share the same computers used by secretarial staff and faculty members.

USES OF COMPUTER EQUIPMENT

Uses for computers vary considerably by department. Word processing is the most frequently reported use with 100% (23) of the departments indicating that function. Data management is the second most important function of the computer with 34.7% (8) departments reporting frequent use and 34.7% (8) reporting occasional use. While only 4.3% (1) of the departments reported frequent use of the computer for grades, 47.8% (11) report occasional use of the computer for this purpose. Problem solving is reported as a frequent use by 26.1% (6) departments with an additional 26.1% (6) departments reporting it as an occasional use. Simulations are frequently used by 13% (3) and occasionally by 34.7% (8) of the departments. Computer assisted instruction is frequently used by 17.3% (4) and occasionally by 39.1% (9) of the departments. Programming the computer is reported as rarely a function by 30.4% (7) and never by (9) of the departments.

FACULTY MEMBERS AND COMPUTERS

Of the 638 faculty members represented, 26.2% (167) have participated in computer workshops or seminars. However, 28.8% (184) can run a commercial program using a microcomputer. Only 13.3% (85) actually assign classroom work requiring use of the computer to students. Other reported uses by faculty members follow:

Administrative tasks	13.7% (88)
Maintenance of grades	8.6% (55)
Programming	7.5% (48)

FUTURE OF MICROCOMPUTERS AT ISU

When asked to project for a five year period, over 80% of the department chairpersons indicated a general increase in use of computers by faculty and students, in the type of uses of their computers, the number of classroom assignments requiring computer use, word processing via computer, administrative tasks, and the amount of equipment available. However, only 47.8% of the chairpersons indicated an increase in the number of faculty members who can program a computer. The chairpersons, then, believe the computer will become increasingly important in their departments.

When asked if faculty being hired for openings should be computer literate, 68.2% (15) of the chairpersons indicated this was a priority. In addition, 90.4% (19) indicated that faculty members should attend seminars, workshops, and classes to improve their expertise in the area of computers. Forty-five percent of the chairpersons believe that the average student in five years will know more than faculty members about computers.

CONCLUSIONS

While the use of computers has certainly increased for administrative tasks, probably because of their time-saving features and while there has been a significant increase from ground zero just a few years ago, the number of faculty members who use computers remains relatively low. Among the reasons for this are the fact that until an individual actually sees a personal profit from spending time learning to use a computer, they probably will not do so. Therefore, if there is no software available that is pertinent, easy to use, and well-documented, faculty may not use computers because of the lack of personal reward. In addition, due to budgetary problems, faculty members are already overloaded and have little, if any time, to spend learning a new skill--regardless of how important that skill is. However, it appears it will become increasingly important for faculty members to be able to use computers for a variety of purposes in the near future. In view of the fact that over two-thirds of the chairpersons indicated they desire new faculty to have computer skills, it may be imperative to use the computer just to obtain employment. The importance of computer literacy for job seekers is not limited to the field of education and, indeed, may even be more critical in other fields.

While seminars and workshops can help faculty members become more computer literate and show them a way in which to use computers effectively, participants must continue to use the skills learned in order to keep them honed. It is easier to enroll in a formal workshop or seminar and attend the required number of hours than it is to motivate self to continue to learn after the workshop or seminar has ended. If universities are to expect faculty to keep abreast of computer technology and programs in their fields, it would seem appropriate to allow released time specifically for improving computer skills. Budgetary limitations, however, will probably never make this possible.

In view of the limitations, it is actually surprising that as many faculty members do use the computers for a variety of tasks.

POLISHING THE APPLE IN VOCATIONAL EDUCATION

By: Jeannie Ireland, Illinois State University, Normal, IL

When microcomputers were first purchased by schools, there was great concern about software availability. Publishers and computer companies quickly identified the profitable market awaiting them and produced a plethora of educational software. Today, there is a great deal of software available for elementary education--especially in the areas of language arts and mathematics. These disciplines also have software available at the secondary level. A glance at almost any software catalog will quickly put educators at ease if they are most concerned about the more traditional subject matter areas. When it comes to vocational education, however, (with the exception of business education) there is a lack of available material.

Several reasons are apparent. FIrst, agriculture, industrial arts, and home economics have lower enrollments than history, language arts, and mathematics for obvious reasons. Therefore, the market is smaller and potentially less profitable. Second, people who program computers are more familiar with the traditional disciplines and can write software appropriate in those areas. When programmers who lack background in the vocational areas write software for these disciplines, they may make errors in fact or in judgment. For example, a commercially available program for industrial arts will accept "purple" for an electronic resistor, but will not accept "violet" which is the proper terminology. Obviously, the programmer was not familiar enough with the technology to interpret the information used. Third, it may be more difficult to produce software for many vocational areas because not every subject has hard and fast rules as in other disciplines. Fourth, vocational areas probably vary more considerably than do the more traditional disciplines when taught by different teachers. While every English teacher must teach certain rules of grammar, one vocational teacher may teach housing construction omitting energy as a consideration. Another may concentrate on energy and make housing construction a more secondary function.

For whatever reasons, there is relatively little software available for the vocational areas. We will review some of that software.

Drafting/Interior Design

APPLE WORLD has a demonstration program on the diskette which illustrates the variety of functions that can be accomplished with its use. Apple World allows users to view objects in three dimensions and from any angle. Users can create anything--a compote dish, a piece of furniture, a room interior, or even an entire house--then look at it not only from a design standpoint, but for functionality. For instance, animation can be used in a room or for a floor plan. Animated characters may walk through a task such as cooking a meal, walk around furniture, or try out traffic patterns. Apple World could be used in businesses or schools for interior design, kitchen planning, furniture design, and to plan spaces for family interaction.

Construction

MECC has several programs designed to help in calculating difficult measurements. For example, one of the programs on their Industrial Arts diskette calculates length of stringers for stairs, adjusts riser and tread measurements, and even lists the board length required to make the cuts. The rafter program on the same diskette calculates rafter length for common and

hip rafters of various pitches.

Means, the firm that publishes the classic construction cost estimating books, now has computer software available to accomplish the same tasks. This software is very sophisticated and even provides a list of materials.

Energy

A problem of major concern to Americans is energy conservation. Since approximately 20 percent of the energy used in the United States is consumed in homes, energy efficiency of homes is a priority for home economists, builders, and home owners and renters. Several programs are available--some of which incorporate solar alternatives.

ENERGY MISER is designed to determine heat loss and heat gain in homes and to compare energy alternatives. Energy Miser can be customized for any geographic location. It calculates annual savings on utility bills for home energy improvements or modifications such as better insulation, weatherstripping, storm windows, and many more. Energy Miser calculates return on investment for energy improvements and includes the energy tax credit. One caution: there is a bug in the program. Entering same figures will not always produce the same results!

ENERGY HOUSE, produced by TIES, is a graphics oriented program that tests awareness of energy usage. A variety of rooms are shown on the screen, each with some energy being wasted. Users type commands to close windows, turn off lights, fix holes, etc. When a correct command is given, the graphics change to fit the situation--i.e. windows close, refrigerator doors close, water heaters are swathed in insulation. If people are in the room, the program will not allow lights, radios, television sets, etc. to be turned off. At the right hand side of the screen is a thermometer as an indicator of the percentage of energy being wasted. Each time an energy waster is eliminated, this indicator changes.

MECC HEAT LOSS requires the use of information obtained from a student worksheet. Questions are asked to determine construction of the house and the R-values of its components. Results show annual cost in dollars of each separate component for heat lost. However, a printer is required to obtain the results. They will not list on the screen.

SOLAR ENERGY not only calculates the heat lost through a house, but additionally permits users to make use of any passive solar energy incident on the area. Thus, sun shining through windows reduces the cost of operation. However, this program will not calculate gains from heat sinks and other passive solar devices. It does make obvious, though, the benefits of sitting a house to use available sunlight.

Finances

A real eye-opener for most people is seeing how much they will actually pay for homes including interest. MORGAG is one of several programs that calculates this cost. Morgag is designed to determine one of four items--length of the mortgage term, the amount borrowed, the interest rate, or the monthly payment. Morgag calculates the desired feature then determines the total interest paid over the life of the mortgage. It will print an amortization table using either annual or monthly figures. The major drawback of this program is that it does not allow users to include insurance and tax data commonly included in mortgage

payments nor does it permit calculation of extra payments on the principle.

There are a variety of packages available that calculate interest, return on investment, and amortization tables. Most operate without a printer and provide valuable information in usable form.

Clothing and Textiles

Computers are used in the pattern industry but little use has been made in the area of clothing construction otherwise. A program listing to create a dress pattern was published in the September 1976 issue of INTERFACE AGE. Gerber now produces a relatively sophisticated pattern generator, but cost is generally beyond the means of school districts.

PERSONAL COMPUTING published "Drapery Estimating" in their September 1980 issue. This program can be had for the cost of keying it in. Pattern size and desired fullness are considered along with length. A program entitled "The Weaver" is used by Bradley University in Peoria. This program allows users to adjust warp and weft yarns to create various patterns, much like a jacquard loom. Users are then graphically shown what the pattern will look like in the finished product.

A figure analysis program was developed for the State Department of Education. Suitable for use in junior and senior high schools, this program asks about personal figure characteristics then suggests the types of necklines, sleeves, lengths, etc. that would be most suitable for the individual.

Food and Nutrition

This is one of the few areas for which a great deal of software exists. There are several sophisticated programs for diet analysis, some of which even consider special dietary needs.

DIET ANALYSIS by Apple requires that users input food and drink for each day. The program then calculates the nutritional content of the foods. A running record can be kept. Although this program will run with only one disk drive, it is more suitable for two disk drives.

MECC FAST FOODS lists several popular restaurant chains such as McDonalds and Kentucky Fried Chicken. An abbreviated menu is provided from which users select foods. Sex and pregnancy status are considered before providing users with the percentages of recommended daily allowances provided by the foods chosen. Users then select an exercise and are graphically shown how long they must participate in that exercise to burn the calories consumed.

KEYBOARDING IN THE K-12 CURRICULUM

By: Truman Jackson, Minnesota Department of Education, New Brighton, MN

The presentation attempted to answer some major questions being raised about keyboarding. Those questions are: Should keyboarding be taught? What is keyboarding/typewriting? At what level should keyboarding be taught: Elementary? Junior High? Senior High? What material is available? What equipment should be used? Who should teach keyboarding?

Should Keyboarding be Taught?
The incorporation of computers into the classroom and the introduction of application software, beginning in elementary grades, makes learning efficient use of the keyboard imperative. As a result, keyboarding instruction has become a part of the elementary school curriculum. However, keyboarding instruction cannot end there, it must continue into the junior high and senior high. A curriculum guide "The Electronic Keyboard for Business and Personal Use", which is available from the Minnesota Curriculum Services Center, describes a recommended keyboarding into typewriting into word processing instructional continuum which has begun to replace the former junior/senior high typing sequence.

What is keyboarding/typewriting?
Keyboarding, simply defined, is learning the correct manipulation of the keys on a computer/typewriter keyboard. Typewriting is the continued development of keying skills and the use of those skills in a variety of applications such as the creation of letters and reports.

At What Level should Keyboarding be Taught?
Keyboarding should be taught in the curriculum at a point prior to using the keyboard for text string entry--(the entering of words and/or letters/numbers such as in word processing, logo, and programming).

In early elementary, the goal of instruction may be to simply aid students in the locating of keys on the keyboard. Beginning at about the third grade level, formal. Development of correct keyboarding technique should be introduced, focusing on the alphabetic keys. At the upper elementary level, a simple to use text editor/word processor should be introduced.

At the junior high level, basic skill development should begin. Also, appropriate beginning applications, such as letters and reports, should be incorporated into the content.

At the senior high level skill development should continue as should applications where emphasis should be placed on office equivalent production standards. The senior high program will conclude in a one semester or one year course in word processing (see "Electronic Keyboarding" for course outlines).

What Materials Should be Used?
Both computer software and textbooks are available for keyboarding instruction. Whereas either set of material can be used, hardware availability and instructor expertise tend to influence choices of material. CAI software for keyboarding has become more sophisticated. Some software (e.g. Microtype The Wonderful World of PAWS) incorporate pedagogically sound

instruction of correct techniques, monitored, paced skill development, frequent speed reports, positive feedback, motivation, student reports and more.

Text material is desirable where typewriters are used for instruction. A combination of software and texts provides more drill material, while providing more variety. Where software and text material drill are used, lines in each should be coordinated (so a student can transfer from computer to typewriter at anytime and continue at the same point in a lesson). Microtype The Wonderful World of PAWS and Elementary Keyboarding from South-Western Publishing Company are examples of coordinated line by line instructional software and text materials.

What Equipment Should be Used?

Either computers or electric/electronic typewriters, or a combination of both, may be used. However, it appears that computers are preferred at the elementary level, then a mix of typewriters and computers at higher grade levels. Manual typewriters are inappropriate, especially at the elementary level. Another important consideration is furniture. Furniture must match student size to assure that table height allows students to reach the keyboard properly.

Who Should Teach Keyboarding?

The instructor should be one who has had an appropriate methods course in keyboarding/typewriting. The methods course for business teachers should aid them in learning to work with elementary age students. Elementary teachers may need to strengthen keyboarding skills prior to, or in addition to learning keyboarding methodology. In conjunction with the creation of Elementary Keyboarding, an outline for two credit methods courses was designed, one for elementary teachers and one for business typewriting teachers, by the task force that helped create Electronic Keyboarding.

How Long Should the Courses Be?

Feedback from on-going programs indicates that approximately 30 hours of instruction at elementary--15 hours each in two consecutive grades or 10 hours each in three consecutive grades, is becoming the most prominent implementation model.

At junior high approximately 60 additional hours of instruction is being offered in grades 7, 8, and 9. The senior high curriculum is one and one-half to two years of additional instruction, including a semester or year of word processing instruction.

THE MICROCOMPUTER LAB OF THE FUTURE

By: Bernie Kelly, Forest View High School, Arlington Heights, IL

Being a prognosticator, in one respect can be very safe, while in another can be very embarrassing. At the time of the forecasting, some may disagree with what is being said but cannot prove the statements wrong. At a later time though, if the statements made did not come to pass and if the person making the statements was remembered, his creditability may be in jeopardy. There is an old saying, "one who speaks through a crystal ball, may one day be eating glass." With these thoughts in mind, I will continue on to discuss my view of the future, primarily, the future of the computer in education and hope that those reading this may forget the author if the statements made do not come true. In order to travel the road into the future, the past must be examined to find any trends that may help to choose the proper path and make the journey a little more realistic.

The use of the computer in education has made three major changes in the past twenty years. In the '60s we taught through the computer; in the '70s we taught about the computer, and the '80s we are teaching with the computer. The computers of the '60s were large, slow, single user, batch processed and expensive. In most schools, access was through a computer center where operators accepted your key-punched cards, put them through the card reader and returned them (hopefully) the next day along with the print out of all of your errors. After several attempts to get that big machine to understand what it was you wanted it to do, you walked away with a ton of paper and a 6 digit answer to your program. At Forest View High School we used an IBM 1620 with 16K memory, a card reader and card punch. The latter was used to create the compiled deck from the authors source deck. Fortran II and IBM machine language were the two means of communicating with the "dinosaur." The output was a 7cps typewriter or punched cards if one liked reading the holes. The purchase price of this machine in 1971 was $50,000.00, used of course! The primary use of the computer in education was to teach programming to a select few, i.e. the computer science students.

The early '70s brought the mini-computer to the education scene. It was faster, had more memory, cheaper, was smaller in size and allowed for several people to be on at one time through what was called a time-shared system. If the computer was nearby in the same building, individual terminals could be "hard wired." If it was not in the same building, access could be accomplished through modems, connected through the telephone. In 1973, High School District 214 purchased a Hewlett-Packard 3000 time-sharing system. The time-share system allowed for multiple access of up to 32 ports. The basic system was cheaper than previous computers but the cost of the terminals and the telephone lines became costly. The computer classes of the '70s were

primarily programming but more more information was being presented about the computer. During this time the computer courses began to grow and some CAI was slowly beginning to creep into classrooms but without widespread success. The cost for multiple user stations was prohibitive. CAI at that time was relegated to drill and practice with very little good software being produced by the software houses.

In the mid '70s the micro enter the schools first in the modular "kit" form. Forest View's first attempt at moving into the micro world was to purchase a SouthWest Texas 6800 Micro-Processor and spent all summer assembling it--solder and test--solder and test. During that same summer, two other people spent their time doing the same types of activities in a garage in California. The following November, we purchased one of theirs. They seemed to have perfected their assembly better than we had. These new pre-assembled computers began to increase in the schools. Since there was very little software written for education at that time, most teachers and students began to program. We switched from the time-share system to programming labs with micros. Enrollment in courses whose major emphasis was programming soared. At Forest View, enrollment peaked at 900 students in 1982 (total school enrollment at that time was 2000).

During the late 70's and early 80's the price of the micro decreased and more software was being written for all levels of learning, from preschool to university. Educators began to talk about enhancing learning with the computer. CAI (Computer Assisted Instruction) software increased in availability but the majority of the material was aimed at drill and practice. The number of courses at the high school increased containing more programming, the history of the computer and social issues of the computer. The microcomputer was now entering it's adolescence stage, growing rapidly, constantly changing direction, and unsure of where it fit into the curriculum and society. It is now reaching young adulthood, and will begin to integrate into the work force and the schools as a working tool; a very powerful tool. Soon it will reach maturity and be accepted for what it is and what it can do on the job, in society and the school. The trend in education is now to teach with the computer. And that can be divided into two ways: 1) to teach <u>with the</u> computer as a tool, word processing, spreadsheets, data bases and 2) to teach <u>with</u> the computer as an aide, drill and practice, simulations, tutorials.

The computer courses designed to teach <u>about</u> the computer in the late '70s contained such topics as the history of the computer and the computer in society. Do we teach the history of the pencil? Do we teach the history of the automobile? Who did invent the automobile? We do teach in the high schools how to use the automobile, how to operate it properly. In some high schools, we even teach how to fix the automobile, which is becoming more difficult without highly technical equipment to help with the analysis of malfunctions.

The microcomputer can be paralleled with the automobile. In the early 1900's when there were very few cars on the streets, the owner needed to know how it worked so that he could make minor adjustments and repairs since there were no service stations or repair garages at that time. In the '30s, '40s and '50s local gas stations were the places to go for repairs. But now open the hood of a new car. You may just close it again--quickly! Many changes have been made under that piece of steel, more than the average owner or gas station

mechanic can handle. The average owner knows how to start the car, operate it and get from one point to another, and not much more. The microcomputer is moving from the stage in which the owner must know what makes it work and into the stage of knowing how to use it. How to make it operate and use the software available to him.

But where do we go from here in education and the use of computers? What do we teach about the computer? What is the course content? How many computers will be enough? How will they be used? Where will they be located? In an attempt to answer these questions, I will use statistics on past and present enrollment and curriculum at Forest View High School and try to take on the role of prognosticator.

At the present time at Forest View, we offer 8 semesters of computer courses in which we teach about the computer. The first course is probably your typical first course, the history of computers, the computer in society, applications, programming and utility programs such as word processing and spreadsheets. This course will disappear from our curriculum in the next year or two. The computer department will refine its courses and the enrollment will drop to a hundred students or so. Students will realize that everyone does not have to be a programmer, but they will need to know how to use the computer and to choose the appropriate software for each situation.

Students will be taught keyboarding and touch typing in the early grades of the elementary schools. Word processing and writing analysis programs will be commonly used at all grade levels. The historical impact of the computer on society, the technical revolution, the information revolution, ethics, copyrights, data banks, privacy, and other social issues will be integrated into future social studies textbooks. These subjects will not belong to the computer science area but to the total curriculum. The curriculum does not need more courses specifically designed around the computer, but, instead, the business education courses must be updated and integrate the computer into all courses the way that it is integrated into the business world. Spreadsheets and data bases will be used constantly in the business education curriculum. The technical and electronics of the computer will be included in physics classes and industrial education courses. The computer will become a tool of many courses and as commonplace as the pencil and notebook. Publishers will be including software with their textbooks which will be integrated to the material contained in the text as they are beginning to do now. Interactive video disc will be used extensively by 1990.

Computer labs will increase in elementary and high schools in the next few years. They will not be used for programming but for CAL (Computer Aided Learning). The term CAL is being used instead of CAI in order to give broader meaning to the use of the computer in education. CAI has in the past been primarily associated with drill and practice. CAL encompasses all educational uses of the computer that enhance learning. In the high schools, there will be microcomputer labs of 25 or 30 units in several subject area resource rooms--math/science, language arts, foreign language, social studies and others where needed.

Why will this all happen? Why will the computer proliferate in the schools? Society will require it. It will not be just a matter of being computer literate to read the newspapers and magazines, but it will be a necessity in order to work in the United States. We will become a nation of clerks. At

the present time, 80% of the work force uses a computer of some type and in the next ten years that will increase. The computer will be as common as the telephone or the car, and word processing on a "lap size" microcomputer will replace the pencil and paper.

Will the computer put people out of work? Will there be high unemployment? There will be unemployment for those that do not have the skills to use the computer and are not able to cope with the information age. The computer will not put people out of work, instead there will be a need for more people in the work force. The need for more people to be employed will increase as the "baby boom" workers begin to retire and are replaced by the workers of the "pill" era. There will not be enough workers to replace all of those that retire. Computers and robots will have to take on added duties and work skills. More automation will be needed. If the telephone industry did not automate the telephone operators jobs in order to handle the amount of calls of today's world, every person, young and old, in the United States today would have to be employed as a telephone operator.

To reinforce some of the statements that I have made, the following chart of Forest View High School, past, present and future, is presented here.

COMPUTERS AND COMPUTER USE AT FOREST VIEW HIGH SCHOOL

	EARLY '60s	LATE '60s to EARLY '70s	MID '70s to LATE '70s	LATE '79s to EARLY '80s	MID '80s to MID '90s
Computer Programming Course Enrollment	15 to 20 students 1/150 (.5-.6%)	30 to 60 students 1/60 (1-2%)	200 to 250 students 1/14 (7-8%)	700 to 900 students 1/3 (28-36%)	60 to 100 students 1/20 (4-6%)
Ratio: Computers to Students	1 / 2500	1 / 300	1 / 140	1 / 30	1 / 6
Computer Course Content	Programming for select few	Programming for select few	Programming for many Adv. Prog. for select few	Computer awareness Programming for majority few Adv Pro	Programming for select few
Computer Use In Other	0%	.01% some Drill & Practice Math & Sci	2-3% Drill&Prac some comp. management Math,Sc,Eng	20-25% Drill&Prac Tutorial as a Tool WP,SS,DB	80-100% as a tool an an aide to enhance learning

TAKING THE NEXT STEP - SLOWLY!

By: Bruce A. Laird, Township High School District 211, Palatine, IL

At a time when the significance of computers in society, and specifically education, continues to rise, secondary school curriculum developers find themselves with few clear-cut, concrete guidelines to follow in assembling a reasonable set of computer courses. Those involved in computer education in District 211 are no exception. The district serves a total of 12,000 students split among five high schools in the Palatine and Schaumburg areas, northwest of Chicago. District 211 has addressed the curriculum situation by updating pre-existing computer courses and methodically developing new computer courses which realistically belong at the secondary level. Offering a potpourri of programming languages to attract a large number of students has been avoided. Computer facilities have been expanded to provide the access necessary at each stage of growth.

Course Descriptions

Two computer courses have been offered for more than a decade. These courses are Introduction to Computers and Data Processing (B222, formerly titled simply Data Processing) and Computer Programming 1 (M224). In the past two years, three new courses have been offered for the first time. These courses are Business Computer Applications (B232), Computer Programming 2 (M234) and Pascal Programming (M354).

Introduction to Computers and Data Processing is taught by the business education departments. This is an introductory course which covers how a computer processes information, potential careers, social issues regarding the use of computers, an introduction to programming using the BASIC language, and applications of computers in the home and in business. The curriculum has been updated extensively over the last decade to stay current with changes in the computer industry. Student use of equipment has moved from exercises with a keypunch machine and unit record equipment to BASIC programming on a timesharing system to the present access to microcomputers. While the course includes some work in writing computer programs, programming is not the main emphasis of this course. Students who are interested in taking this course are encouraged to do so before completing any of the other computer courses. There is no prerequisite, however completion of at least one semester of typing is recommended. This is a one semester course and may be taken in grade 10, 11, or 12. The text used is Introduction to Computers and Data Processing by Shelley and Cashman from Anaheim Publishing Co.

Business Computer Applications is a second course in computers and business which is being offered for the first time during the 1984-85 school year. This course expands on topics introduced in B222. Students gain experience with business applications through class discussion and hands-on use of commercially available programs. Students learn to utilize and process information using word processing, data base and spreadsheet programs. Specifically, the AppleWorks integrated software package from Apple Computer, Inc. is used for the hands-on work. The course also covers file organization and provides an introduction to COBOL programming. Again, computer programming is not the major emphasis of this course. B222 is a prerequisite for this applications course. Business Computer Applications is one semester long and may be taken in grade 10, 11, or 12. The text used is Computers Today by Sanders from McGraw-Hill Publishing Co.

Computer Programming 1 is an introductory course in computer programming techniques. Students learn to analyze problems, formulate solutions to those problems and implement the solutions in the BASIC programming language. In the process, students are expected to write a number of original programs and enter and test the programs on the computer. Students learn the same fundamentals of BASIC taught in B222 but learn some advanced techniques as well. All topics are covered in greater depth than in B222. Revisions in this course over the years have involved updating course materials in light of changes in the computer hardware available for student use. A grade of C or better in Algebra 1 is a prerequisite for this course. A semester of typing is recommended. M222 is one semester long and may be taken in grades 10, 11, or 12. The text used is Computer Programming in BASIC by Golden from Harcourt Brace Jovanovich.

Computer Programming 2 expands on the programming techniques learned in M224. Content includes advanced design and programming techniques with continued use of the BASIC language. Students again write original programs and test them on the computer. Emphasis is placed on the use of subroutines and the organization and application of data file storage. M224 is a prerequisite for this course. M234 is one semester long and may be taken in grade 10, 11, or 12. No text is used with this course. The Apple II User's Guide by Poole, McNiff and Cook from Osborne/McGraw-Hill is used as a reference manual.

Pascal Programming is an advanced course covering a number of topics in computer science. Structured programming concepts, various data structures and commonly used programming algorithms are discussed and implemented using the Pascal programming language. Students write original programs in Pascal and test and debug them at the computer. The course is not taught as an advanced placement course in order to appeal to a broader group of students. However, supplementary materials are available for students desiring independent study on additional topics if the student plans to take the Advanced Placement examination in computer science. A grade of C or higher in M234 as well as completion of or concurrent enrollment in Algebra 2 is required as a prerequisite. M354 is one year long and may be taken in grade 11 or 12. The text used is Pascal by Dale and Orshalick from D. C. Heath and Company.

Recommended Computer Course Sequences

Four sequences of the available computer courses are recommended to students, depending on their area of interest and the extent of their interest in computers.

Sequence 1 is recommended for students who are definitely interested in the field of business data processing or computer science. These students should take all of the available courses in a logical order. The three year sequence of courses would be B222, B232, M224, M234, and M354. This will also require the completion of two years of algebra and one year of geometry.

Sequence 2 is recommended for students interested in the computer as a problem-solving tool in some discipline other than business or computer science (e.g. engineering, mathematics, science, etc.). These students should complete two years of algebra and one year of geometry.

Sequence 3 is recommended for students who are interested solely in the use and application of computers in business. These students should complete B222 and B232.

Sequence 4 is recommended for students who are unsure of their interest in the computer and data processing fields, but who want to learn about these fields. These students are encouraged to take B222 as an overview to clarify their degree of interest. They should then be better able to decide whether to take additional computer courses.

Computer Facilities and Enrollments

The initial move to use microcomputers in computer courses took place in 1981. Twenty microcomputers, equally distributed among the five high schools, served a total enrollment of 969 students in B222 and M224. This was the year in which enrollments in computer courses in District 211 began to climb steeply after being relatively stable at approximately 400 students for at least three years. The following year (1982), a laboratory of sixteen networked micros in each school was shared by 1,399 district students in the same two courses as the previous year. M234 was introduced in 1983. That year found 2,641 students enrolled in the three computer courses. With no increase in available equipment, computer time was at a premium. With two new courses (B232 and M354) to be added in 1984, plans were developed to expand the computer facilities. A second laboratory of sixteen microcomputers in each school with dual floppy disk drives and CP/M cards serves the needs of B222 and B232 students, while M224, M234, and M354 students make use of the networked lab. A total of 3,078 students are enrolled in these five courses this year across the district.

Computer Languages: Pros and Cons

A variety of programming languages have become the subject of computer courses at the secondary level, especially with the advent of microcomputers. These languages include BASIC, FORTRAN, COBOL, Pascal, and assembly language. The motivation for including such courses in the curriculum varies. The appropriateness of offering complete, in-depth courses in all of these languages at the high school level is questionable. A computer survey was conducted by this author in 1983. Responses were received from 51 colleges and universities. The data submitted by 40 computer science departments, 26 mathematics departments, 41 science departments, 15 engineering departments, 29 business administration departments, and 22 education departments is referenced here.

BASIC is and has been the mainstay of computer courses in high schools since the late 1960s. Early users worked through terminals on small minicomputers, later moving up to timesharing systems. The 80s found microcomputers as equally affordable as timesharing terminals, but having greater capabilities. The resident BASIC interpreter of the micros makes this the language of the masses. Regardless of age, those people who wish to learn a programming language find BASIC the most accessible alternative, both at school and at home. Consequently, BASIC is a logical choice for schools to use as an introductory language, both for students who only wish to "get their feet wet" and those who plan to continue their computer training. The survey of colleges mentioned earlier found BASIC to be the programming language most often taught first to students majoring in five of the six departments summarized, engineering being the exception. However, BASIC is not without its detractors. One such detractor expresses concern about the effect

learning "street BASIC" has on students in their future work with other programming languages. While many BASIC interpreters do not require the use of structured programming techniques, any instructor wishing to teach emphasizing this approach will find few, if any, barriers. Count on seeing BASIC occupying its present niche for some time to come.

FORTRAN is a language that was entrenched as an introductory language in colleges and universities before the development of BASIC. In fact, student difficulties with FORTRAN were the reason Kemeny and Kurtz developed BASIC in 1964. The present day popularity of FORTRAN is now more limited. Most use seems restricted to the engineering and science areas, as indicated in the survey. FORTRAN, like BASIC, lacks clean implementation of structured programming constructs. Also, access to FORTRAN on a microcomputer generally requires the purchase of additional software, and perhaps hardware, for implementation. The limited subsequent usefulness of FORTRAN to most high school students should put it low on the list of languages to be taught.

COBOL commands recognition for its impact in the computer industry. There is no doubt that COBOL has and will continue to have great significance in the field of business. Even with the talk of the federal government's intentions to scrap COBOL as its standard programming language, the extent to which COBOL is currently used will mean that many COBOL programmers will be needed to at least handle maintenance of existing programs. One of COBOL's strong points is certainly its structured programming environment. However, the suitability of COBOL as a language to be taught in the high schools is questionable, not because of the language itself but rather due to the students. Its downfall comes from the fact that most high school students lack sufficient background in business to adequately understand the usefulness of the language in a business environment. Solving business problems is difficult in any language without some background in business. Unfortunately, COBOL cannot easily be used for problem solving in other fields. Thus, students and teachers are likely to be stymied in trying to effectively apply that which is to be learned. Often, the course boils down to simply learning and using the COBOL syntax, perhaps with little understanding of significant applications. Implementations of COBOL on microcomputers do not improve the situation at all. A brief exposure to COBOL with simple programs is appropriate as long as both the students and the teacher accept it for what it is.

Pascal is a relative newcomer at the high school level. That has not restricted its popularity. The language has much to recommend it in the way in which programs are structured. Further impetus for the teaching of Pascal has undoubtedly come from the College Entrance Examination Board (CEEB). The commonplace use of Pascal in beginning computer science courses in colleges and the selection of Pascal by the CEEB as the language to be used in the Advanced Placement Examination in computer science make Pascal a logical choice for inclusion in the high school curriculum. This selection by the CEEB came at a time when schools felt a need to expand beyond BASIC but saw little compelling evidence to support the selection of FORTRAN or COBOL. Over 4,000 students took the AP computer science exam in its first year, 1984. Some 10,000 are expected to take the exam in 1985. In the college survey conducted by this author, 22 of 40 computer science departments indicated that credit would be granted for an acceptable score on the AP test, but only 1 of the 40 institutions planned to allow advanced placement in the computer science curriculum. A recent article in Infoworld (November 5, 1984) reported similar results. With time, this situation should improve, despite the

holdouts among the top institutions for computer science. The importance of Pascal for college-bound students interested in computer-related fields will continue to increase.

Assembly language fails to meet any significant needs of high school students. A student's personal interest may draw them to such a course (the hacker mentality). However, students seeking employment after high school will find few, if any, job opportunities using assembly language which doesn't require additional training in the language. College bound students will generally not encounter assembly language in their early course work in computer science. Only 1 of 40 college survey respondents from computer science departments indicated assembly language was taught in the first course taken by computer science majors. BASIC (50%) and Pascal (42.5%) lead the field in this area. Just 55% (22 of 40) reported that assembly language was taught at all. It seems assembly language will be used by so few, that such a course is better left to the community colleges, colleges and universities.

Finally, a general comment on offering programming languages at the high school level. A major concern when considering which languages to teach is staffing. The best laid plans quickly disintegrate when one realizes that a magnificently planned course which appeals to a great number of students cannot be offered because there is no teacher available with adequate training to conduct the course. Many BASIC teachers are self-taught and have honed their skills on the job. However, significantly fewer teachers have any experience or training in the other languages mentioned. Few colleges and universities have offered computer education programs to train teachers for such courses. Those teachers with extensive computer training are often lured away by business and industry. If the high school course is offered with inadequately trained staff, the consequences can be far reaching.

Summary

As Snoopy once said, "There is no bigger burden than having a great potential." The potential for a quality computer curriculum at the secondary level needs to be carefully cultivated. In the rush to bring computer technology to the schools, we must not forget to guarantee quality before quantity. Hopefully, the ideas presented here will help those responsible for planning computer curriculum in their schools do just that.

CAI—IS THAT ALL THERE IS?

By: Marlene Lazzara and Judy Parker, Henking School, Glenview, IL

During the three years that we have been in the business of using computers with elementary school children, our ideas have changed greatly. We began as most schools do, with a couple of Apple computers and not much more than that in software. Being novices in the field, we learned right along with our students. The computers were used daily for CAI. Then we discovered BASIC programming! What a new world this opened up. Our fifth and sixth graders were clamoring to get to the keyboards, and, as usual, we learned right along with our students. Most days we were lucky to stay just one step ahead. We rolled along quite comfortably for awhile until we found LOGO. This we introduced to our fourth grade students and were somewhat pleased with the results of our efforts. (The sixth graders wanted to know, "How did they do that!". And we were often heard to say, "No commands over 1000, please!").

By now, we were up to four computers in our Learning Center and CAI was still very much in the picture. Our kindergarten through third grade classes did nothing else. During this time, we tried to make a practice of choosing software that we felt was the most valuable for our needs and was of the best quality. Our intermediate students--fourth through sixth grades--still used many of these programs and often alternated programming with CAI. With the addition of roll-about carts, we tried to encourage the use of the computers in the classroom when they were not being used in the LC.

During our second year, our district clustered and we became a primary building. We decided that this was the time to take a new direction in the Learning Center. Our area of emphasis became problem-solving. Only those programs that would encourage and develop thinking skills would be used in the LC. Our old stand-by, drill and practice programs, could now be used in the classroom whenever the teacher requested them. We found that the computer moved out of the LC more often, but not as often as we had hoped. We began teaching LOGO to our third graders, but, of course, had to move much more slowly.

During the course of those two years, one goal remained the same: to have each and every one of our students on the computer from twenty to thirty minutes per week, depending on grade level, for the entire school year. We were very successful in this respect. Students, teachers, and parents were quite pleased with the overall program. Why, you may ask, were we still not satisfied?

As we became more at ease with our computer knowledge, we also began to realize that the public at large was becoming more sophisticated. Many of our students were now the proud owners of computers; moreover, they were using many of the same programs at home that we were using at school. We realized the novelty of using the computer was no longer a valid reason to warrant taking students out of classrooms. We wanted these computers to be used more creatively, to support and enhance the learning that was going on in our building. CAI, however, still lived up to some very realistic needs. It was another useful tool for many students to get needed help in some curricular areas.

During this past summer our building planning committee addressed the issue of computer use and in September we came back to school looking forward to using our computers as tools to complement the learning process. The primary thrust of our teaching was to incorporate language experience in all facets of the learning process. Our staff was introduced to flexible scheduling in the Learning Center. This allows the LC staff to plan with each teacher weekly the kinds of learning activities that will utilize a variety of media. The LC has become the hub for all learning that takes place in the building. Computers fit very nicely into this new philosophy of using that tool which best suits individual learning styles. With the many "teacher utility," "authoring language," and "shell" programs we have available to us, we can now meet most of the needs of the classroom teacher.

Programs that encourage thinking and problem solving still are often used. These are programs such as GERTRUDE'S PUZZLES, ROCKY'S BOOTS, LAYER CAKE, and BOXED IN, that allow the student to work by himself or with a partner to discover that there is more than one way to attack and solve a problem. LOGO is still a big contender when it comes to the problem solving department, and we have experimented somewhat with Houghton Mifflin's Introducing LOGO To Primary Children" and have been pleased with the results.

With our new goals of using the computer for CAI in the classroom and strengthening language experience activities in the LC, we have developed our present program. CAI is being used, but much less often than before. Getting the computer into the classroom is still being met with some resistance. Many teachers do not feel comfortable with the computer in their room. Some feel the computer is a distraction while other learning is taking place. Only occasionally will a teacher request a drill-and-practice program for a particular student or group of students and ask to have the computer and the appropriate software sent to the classroom. A few have requested that we send one computer to their classroom for an hour or an afternoon on a weekly basis. These teachers do make a point to request specific software (or at least, a specific type of program).

One of our greatest areas of success has been that of using the computer with language experience activities. It is here that most of our teachers have shown the greatest interest and support. Our kindergarten teachers have invited us into their classrooms to listen as their students tell us about a field trip from which they have just returned, or to retell a nursery rhyme or fairy tale. As they dictate, we use a word processing package such as APPLEWORKS to enter the story into the computer. When they have finished, we print out a copy of the story for each student. These are usually illustrated before being sent home to parents. Teachers are enthusiastic since this presents the students with immediate reinforcement as they relate what happened on the field trip or during the story, and helps develop sequencing, verbalization, grammar, and listening skills.

Our first grade teachers spent the first few weeks of school discussing rainbows. Together we planned activities to extend this unit, using print and non-print materials. As a culminating activity, students came to the Learning Center to hear the poem Hailstones and Halibut Bones and then were asked to create their own poem about the colors of the rainbow. Discussion was led by the classroom teacher while we input their responses. This took several days, (sometimes we went into the classroom and sometimes they came into the

Learning Center), but the students created a beautiful poem which they titled, Our Rainbow Book. Many new adjectives were learned as the children tried to describe the precise image they had in mind, e.g., "Red is juicy, plump, delicious apples." We printed a copy for each student and had them illustrate the poem before being bound as a book to take home.

One of our first grade teachers culminated a unit on bees by having the children make up an experience story about what they had learned. They then came to the Learning Center to dictate their story. We typed it exactly as they dictated and, using our Nova Beam, projected this onto the movie screen. The teacher went over each sentence with them to see if they could improve their grammar and if their facts were accurate. As they "edited" their story we corrected it. When we were finished, each student had an accurate account of what they had learned about bees without having to go through the laborious (especially for first graders) task of rewriting.

We have heard many "stories" since then: pumpkins and monsters and a paraphrasing of the "Twelve Days of Christmas" were fun for us to listen to and convinced us that children certainly do have beautiful, vivid imaginations when allowed to express themselves freely.

We realized, however, that while we were all pleased with the results of this language experience activity, the students weren't having any "hands on" computer time. So, we tried letting some students type their own stories. This was a disaster! In fact, it was easier for them to write and rewrite than to try to input into the computer. Thus, our emphasis in the LC has become that of keyboarding and word processing. All of our second and third grade classes have begun keyboarding. Because of time constraints, only two of our classes come in every day for fifteen minutes per group. The other second and third graders are in once a week. A keyboarding session begins by booting up with BANK STREET WRITER and using a keyboarding booklet developed by a typing teacher at our junior high. (We have previewed several keyboarding programs and found none that met our needs. We are waiting for the arrival of two new programs designed for students in kindergarten through second grade.)

In the beginning, we found our students were more concerned with correcting errors than learning proper fingering. We quickly found that if we turned the monitors off, fingering and eyes-on-the-book soon became the major concern. After a student finishes a page, he calls one of us over to turn on the monitor and discuss what errors (if any!) have occurred. We now save our daily work on diskettes to be printed out so that each student may have a copy of what was accomplished during class to take home. This has necessitated learning how to use BANK STREET WRITER.

In addition, we have color-coded all our computer keys with the hope that our kindergarten and first grade children will begin proper fingering. When the children come in to use the computer, we place a small dot of crayon on each finger to correlate with the dot on the computer. This encourages the use of both hands and also proper fingering. Students are encouraged to rest their fingers on the home row when at the computer. Worksheets have been used to reinforce specific finger areas of the keyboard.

We have come to believe that the most valuable use of the computer for students is in word processing. This is the one area that no other medium can approximate. We have many methods to allow students drill and practice (which does not teach--merely reinforces). Our problem solving software is excellent, but we cannot rely solely on these since computer time is limited. Teaching students keyboarding and to use a word processing package frees them from laborious editing and allows them to experiment with language. We are finding that primary aged children are not too young to begin learning these skills; the genuine enthusiasm and creative thinking we see which enhances the learning process makes our efforts worthwhile.

OUTLINING A PRACTICAL FIRST COMPUTER COURSE FOR SPECIAL EDUCATORS

By: Kent and Barbara Luetke-Stahlman, Northern IL University, De Kalb, IL

In the summer of 1984 the authors taught a graduate level course, "Computer Literacy and Software for Special Educators", to 24 participants at the University of Nebraska, Omaha (UNO) campus. Participants had little or no prior knowledge of computers and were either teachers of Behaviorally Impaired, Learning Disabled, or Hearing Impaired students or Speech and Language Pathologists. The course met twice a week for three hours each session over a five week period and was team taught by the authors. Barbara Luetke-Stahlman was the coordinator of the UNO Hearing Impairment Program, receiving her doctorate in that field from Pennsylvania State University in 1982. Kent Luetke-Stahlman was a computer science teacher and coordinator for District 66 in Omaha. A large part of his job involved organizing a computer center at the high school and inservicing teachers on computer operation and software selection.

Due to the differing expertise of the instructors, coupled with the desire to allow for maximum individual computer time, participants were split into two groups for discussion of several class topics. In Table 1, these smaller sessions are labeled as Groups a and b. Using these groups, the instructors would present their topic information, rotate the groups, and repeated their presentation.

The class regularly met at the Educational Computer Lab at UNO. Here, 24 students shared 12 Apple II+ or Apple IIe computers when "large group" occurred. There were only two printers in this lab. At UNO, students could work individually when divided into groups, although they occasionally still choose to work together. Twice during the course, students met at the District 66 high school computer center. This facility afforded each student individual access to either an Apple II+ and printer or Apple IIe computer and printer. Hagen (1984) Microcomputer Resource Book for Special Education was suggested as a text and referenced in lectures, but there was no assigned reading in the course.

THE COURSE

The topics included in the course were word processing, five types of software, I.E.P. evaluation, Logo and management systems.

WORD PROCESSING

The session on word processing introduced the participants to the difference between a home limited featured word processor (i.e. Bank Street Writer's first version) and a full featured word processor (i.e. Applewriter). Both kinds of word processors were discussed in a five part evaluation involving 1) Enter text; 2) Edit text; 3) Store text; 4) Print text; and 5) Spelling correctors. A detailed hands-on session using Bank Street Writer followed such that participants were shown all the features of Bank Street Writer. Students were then asked to write two double-spaced paragraphs--requiring the use of the return key and access of the transfer menu. This assignment gave participants that great sense of joy seeing their first word processing text roll off the printer.

FIVE TYPES OF SOFTWARE

The authors have divided the instructional strategies that a microcomputer can be used for into five categories. Drill and practice, reinforcing skills at the lowest level of the student learning hierarchy, uses the computer's capacity to be patiently repetitious. Simulation (which requires non-computer time to insure students know and comprehend the information involved) is used to explore alternative solutions to problems in a user-controlled environment. Tutorial software can introduce new material and interact with the student according to the individual student's need. Problem solving software allows the student to manipulate "what if" situations and to explore new solutions to the same problem. Educational games take all the advantages of the above four categories but allow the teacher to interact in an authoring mode to choose individual goals and materials. In Table 1 is a list of software that the participants then were given time to review.

I.E.P. EVALUATION

Commercial I.E.P. packages have been flooding the market in the last year. Our students were exposed to five such packages, sent to us for demonstration purposes by the distributors. Some of these programs included a limited set of objectives with no authoring capabilities while others allow professionals to create their own. Use of a limited bank of objectives causes one to pause and question the original intent behind the 1975 federal law which mandated individualized educational programs. Professionals using these packages need to consider the authoring capabilities and the completeness of the objectives which can be generated. Many programs use weakly written objectives, containing verbs describing unobservable behaviors of omitting condition statements. Costs of I.E.P. programs do not necessarily reflect these features. A recommended program is by Parrot Company (Weiner, 1984) which costs about $50.00.

LOGO

Using Watt's (1983) Learning with Logo as a reference, a lecture format was utilized to give participants Logo commands. They were then asked to type them in and explore them. Next participants explored a complete project (stick figures p. 109-113), which lead to the assignment of constructing a truck. Participants were given circle procedures for wheel on their Logo diskettes and were required to read them into their program. Saving their trucks on diskettes required use of the editor. Finally, printing the trucks out on a MX-80 Epson with a Grappler interface required students to follow a sequence of steps before printing.

MANAGEMENT SYSTEMS

Both commercial gradebook programs and data bases programs were discussed. Three general terms (i.e. file, record and field) were given to help understand the organization of data base. The concept of the computer as an electronic filing cabinet, a drawer of the cabinet as the file, an individual folder as a record, and the individual entries in a folder as fields were utilized. Participants were then taught to use three components of a data base program (i.e. sorting, searching, and listing).

EVALUATING STUDENTS

The grading system utilized for the course was based on two quizzes, four demonstrated competencies, and a final class project. That project was assigned as follows:

> Each student will be expected to work with at least one other student in our course to compile the resources for one of the handicapping conditions listed below. Your grade will be based on (1) the organization of your presentation; (2) the comprehensiveness of your class handout; (3) your ability to work as a team; and (4) your use of word processing software to generate your handout.

The projects were entitled "Computer Resources for Developmentally Disabled Students" (Wiehl, Brawner, Hutchings, Whalen, and Peterson); "Computer Usage and B.I. Students" (Vernon and Vaughn); "Learning Disabilities" (Barnhizer, Smith, Webb, and Heckathorn); and "Speech-Language Computer Resource Book" (Keith, Kladde, Miller, Reed, Rucksdahsel, and Tompkins). All are available for the cost of copying and postage from the authors. The project entitled "A List of Computer Software and Relevant Information Dealing with the Hearing Impaired Population" (Weiss, Blume, and Reedy) has been accepted for publication by Perspectives (Gallaudet College, 1983) and will appear in 1985.

TABLE 1. SUGGESTED SOFTWARE FOR SPECIAL EDUCATORS

COMPUTER LITERACY

1. The Friendly Computer — MECC

MATHEMATICS

1. Arcademic Skill Builders Math — DLM
   - Alien Addition
   - Minus Mission
   - Meteor Multiplication
   - Demolition Division
   - Alligator Mix
   - Dragon Mix
2. Basic Number Facts — Plato
3. Metric Skills I & II — Hartley
4. Expanded Notation — Hartley
5. Clock — Hartley
6. Algebra Arcade — WEPCO
7. Bumble Plot — The Learning Co.

PROBLEM SOLVING

1. Moptown Hotel — The Learning Co.
2. Moptown Parade — The Learning Co.
3. Rocky's Boots — The Learning Co.
4. Gertrude's Puzzles — The Learning Co.
5. The Factory — Sunburst
6. Baffles — Conduit
7. Snooper Troops #2 — Spinnaker

LANGUAGE ARTS

1. Word Invasion — DLM
2. Verb Viper — DLM
3. Spelling Wiz — DLM
4. Capitalization — Hartley
5. Word Families — Hartley
6. Vowels — Hartley
7. The Stickybear ABC — Xerox

AUTHORING SOFTWARE

1. The Game Show (Computer Advanced Ideas)
2. Micro-Interpreter 1 (Microtech)
3. Word Analogy (SEI)
4. KinderComp (Spinnaker)
5. Story Machine (Spinnaker)
6. Facemaker (Spinnaker)
7. Delta Drawing (Spinnaker)
8. Keyboarding (Plato)
9. Blissymbolics (MECC)

READING

1. Readability Index (Educational Activities)

COLLEGE CHEMISTRY GRADUATES HAVE POOR COMMUNICATION SKILLS: WHAT TO DO?

By: Peter Lykos, Illinois Institute of Technology, Chicago, IL

Standards for Bachelor's Degree programs in chemistry are set by the American Chemical Society Committee for Professional Training (CPT). Of the 11,000 Bachelor's Degrees given every year in chemistry, 10,000 are from the 575 departments which have been approved by CPT. Chemistry has been the only scientific discipline (including math) to have such an approval program in place until recently. The ACS/CPT Approval Program moved the ACM (and subsequently to invite the IEEE-CS to join forces) to create the Computer Science Accreditation Board last May. We can expect that some externally imposed standards will be coming to Bachelor's Degree programs in computer science as well. The ACS/CPT frequently samples the hirers of Bachelor's Degree holders in chemistry in order to learn how the guidelines for approval should be revised in order to better prepare students for a career in chemistry within ACS/CPT approved programs. The major and most consistently appearing suggestion for improvement of the 'product' has nothing to do with chemistry per se but with the ability of graduates to communicate. At its annual retreat meeting last December, ACS/CPT included a multi-part briefing for the 12 CPT members which lasted for almost an entire day regarding ways in which communication skills of undergraduates might be improved without adding to the number of semester hours in the curriculum.

Three different approaches were considered:

(1) TOASTMASTERS INTERNATIONAL is a mutual help organization for persons who have finished school (usually college) and who have found that an essential barrier to realization of their potential is their poor speaking ability. The problem is a common problem--several for-profit organizations have become quite successful conducting intensive course and training programs which are expensive. TI has 5,000 clubs around the US which take advantage of many years of experience in the form of printed guides and successful products of the system to help motivate persons to improve their speaking skills as well as their skill at running meetings. Chemistry departments could incorporate such an approach into the departmental course, seminar and research group meeting structure.

(2) THE WRITERS WORKBENCH provides a machine-based text analyzer for critiquing written text. The publicity given to the Writers Workbench has led to the creation of many derivative systems. A problem with the original Writers Workbench is that it operates under the UNIX operating system which is not yet in widespread use (particularly on personal computers) and that it requires a screen editor interfaced to the UNIX/WWB combination. It also does not do grammar analysis. Derivative systems are now available which have the following features:

(1) Run under MS-DOS and on IBM PCs and their clones.
(2) Interface with common word processors, such as Wordstar.
(3) Go beyond spelling checking and elements of style and readability level analysis to deal with some aspects of grammar as well.
(4) Cost less than $100.00.

(3) In addition, a team of two chemistry professors from UCLA described a company they formed and which has become quite successful in providing in-house training programs for chemical industries. Their recommendation to CPT is that a series of workshops be established in connection with the semiannual ACS National Meetings whereby chemistry faculty could be taught some of the techniques they have found to be effective in teaching chemists how to improve their writing skills.

The first two approaches have the benefit that the supporting infrastructure is already in place while the third could evolve only gradually.

Another consideration is the so-called coop approach. Students who co-op spend five years working on their Bachelor's Degree during which time they spend four one-semester periods or four six-month periods working in a chemical industry. Their supervisors stress communication skills and work with the students to help them improve. In addition, their supervisors work on the student's social skills--the ability to get along with others is also high on the job requirements list.

The problem is real and severe in intensity. Remediation programs which colleges have had to put in place over the past several years are not sufficient. Indeed, the Association of American Colleges created a project on Redefining the Meaning and Purpose of Bachalaureate Degrees and the corresponding committee published its report last month, Integrity in the College Curriculum: A Report to the Academic Community. In that report a minimum required program of study was delineated which included nine points. The second point was identified as Literacy: Writing, Reading, Speaking and Listening. Articulation with the secondary school failure was highlighted as well as a condemnation of teachers colleges for turning loose on the nation tens of thousands of inadequately prepared persons.

The day has arrived when professional accreditation bodies for the sciences need to deal with lack in basic skills on the part of their 'feedstock' which in the recent past could be presumed to be part of their secondary school education.

Chemistry professors teaching English?

COMPUTER POWER: A PROBLEM SOLVING COURSE FOR STUDENTS OF ALL LEVELS

By: John W. McConnell and Patricia Finegan, Glenbrook South High School, Glenview, IL

We are not among the alarmists who view BASIC as a corrupter of American youth. That language was designed for beginners; novices can do some very pretty things with it; it is built into most microcomputers; and astute teachers can still promote good programming habits with it. As an interpreted language, users can trap errors more easily than in compiled languages.

Some of the problems in the teaching of programming courses which have been credited to the BASIC language are due more to the content of introductory courses than to a specific language. Typically an introductory course builds from the primitives of the language. Initial problems which students can solve in the language are necessarily trivial. If we want to establish computing as an area of curriculum which should have impact for all students, we must look at introductory experiences which feature the computers as an intellectual tool. In this paper we argue for the adoption of an introductory course which looks at the broader benefits of computing and which is commercially available. That course is Computer Power. It is a course which we have found to be exciting, stimulating, and stunning when we have used it with real, live heterogeneous adolescents.

We believe that one question in particular should motivate curricula in computer programming for all students:

What is the role of computer experience in the liberal arts education?

We ask this question because we believe that the answer will provide the most durable justification for computer education in our schools. A firm rationale for computer courses targeted to the majority of students is necessary because many popular contemporary arguments used for programming will fall by the wayside in the immediate future. Here are some of the reasons educators use to justify courses in BASIC and why they will disappear.

> Jobs - The vocational goals of many programming courses will disappear as high level languages will remove programmers as intermediary between user and machine. Programmers are more likely to be in over--supply a decade hence than under--supply.
>
> Literacy - What should students know about computers? The study of the effects of technology on society will not sustain programming courses, but may influence the content of social studies courses.
>
> Common, Generalizable Language - Even though all popular microcomputers speak BASIC, most buyers acquire software which help them solve problems and which never require them to learn BASIC. There are languages, such as Pascal, which are becoming as common as BASIC and have structures which will be more generalizable over machines and language implementations.

Computer courses aimed at the average high school student should be tested against their capacity to extend the thinking power and problem solving skills of students. That is what we mean by the liberal arts rationale for programming courses. A course which is built on that rationale is Gregg/McGraw-Hill's Computer Power. We quote liberally from the preface of the Teacher's Manual.

> We have observed that most computing courses, whether in high school or college, have the effect of driving away a large group of students, those without strong abstraction and verbal skills. These students are often quite intelligent and capable, but their minds do not easily learn when the environment deprives them of their favorite kinds of information. Many of these students "come alive" in this course, because the course is very carefully structured to provide visual information, color, and sound at every step. Perhaps "computer anxiety" is not the appropriate term for these students' antipathy to linearsymbolic logic. A better term would be directed at the curriculum's deficiencies rather than the kids': we could speak of visually deprived curricula. This project has been motivated by a need to avoid visual deprivation.
>
> Using the visual domain as a "hook" to keep students interested, we then pursue those skills of abstraction which are needed for problem solving of all kinds. Computer programming requires the ability to plan, to dissect problems into smaller problems, to image things you cannot see, to correct initial hypotheses as they are proven wrong. All of these skills contribute to problem solving ability in any domain.
>
> The course is fundamentally a programming course. Even though programming skills are not the most important goal (we think problem solving is more important), programming is easier to get hold of, to teach, and to evaluate. Programming must be taught correctly, or many bad habits can be learned. This curriculum was created by people involved in the mainstream of the "structured programming" movement within computer science. The programming techniques in this course are very up-to-date, and in fact many universities do not yet use them all. Your students will be able to enter computing courses anywhere with a superior foundation, giving them a significant head start on their further education and careers.

While we have cited the problem solving aspect as important. the preface of Computer Power defined two other areas which we consider consistent with the liberal arts foundation of programming. First, the authors have recognized the paucity of visual communication in the high school. Our students must be able to go beyond the printed page. Schools are trapped in the Gutenberg age - if it can't be typeset, then it doesn't exist. Contemporary society cries out for a broader base of communication. The computer provides an interactive, visual medium which will elevate pictures and animation to the station of words as carriers of human thought. Second, the aim of the authors of Computer Power to teach good programming habits is independent of the vocational goal: the program is viewed as an embodiment of a solution to a problem. It is a communication, a crystallization of human thought.

An introductory computer course should feature problems of substance, provide a language rich in structures which contribute to the solution of problems, use the graphics capability of the computer, link to other work in programming, and show students results and errors quickly. The creators of Computer Power have done just that.

1. Problems. What problems are relevant to the average grade 8/9/10 student? Computer Power features cartooning. We have taught a mixture of students in our courses. Some have had some computer experience and abilities have ranged from special education level to genius. We were skeptical that cartooning could provide all students with a challenge. It does. The problems presented to the beginning student in BASIC are trivial compared to the problems presented to the beginning student in Computer Power. The adolescent is far away from mathematical and data analysis problems which form the basis for most programming problems. Their experience is tightly linked to cartooning.

2. Language. The developers of Computer Power worked with UCSD Pascal. Their variant, "Rascal," has a multitude of virtues. Among them is a rich variety of procedures for cartooning. Procedures of MOVE, SPIN, FLY are constructs which link naturally to animation. Pre-drawn figures (actors) can be called from disk (Jack Coustoad, Patricia Pig). Instead of working from primitives (ala BASIC), the students have a language which is close to the problem. The visual geometry of movement, symmetry, and transformations matches the experience of the students and is immediately accessible in the language.

3. Ease. UCSD neophytes grimace over their early exposure to the p-system. Computer Power introduces students to the program editor with great grace. Poor typists can call up procedures with single strokes. Execution of the program is done within the program editor. Rascal is an interpreted Pascal. There is no waiting for the program to compile. At the early stages of the course, the Rascal line being executed appears on the screen while the cartoon action is performed, so errors in logic or choice of variables are easy to trap.

4. Connectionto other programming. Computer Power is a semester course. During the summer school session in which this was taught at Glenbrook South, we covered the main topics of the course in 10 four-hour sessions. We then devoted 5 four-hour sessions to UCSD Pascal. Students were able to complete a Pascal graphics project within that time. The move to the full UCSD system was exceptionally easy to make.

In addition to the virtues listed above, Computer Power has been designed as one of the most realistic instructional packages available on the market today. In our opinion, it sets the standards against which we should measure other introductory courses.

1. Exploration. Every lesson is designed to take 5 days. The initial two days of a lesson require that the student explore features of the language or system. This encourages students to formulate questions and hypotheses. The goal is to encourage the student to absorb the meta-message, "You can do this stuff as well and as beautifully as anyone else."

2. Group Work. The course is designed for classes which are rich in students but poor in computers. Exercises are designed for group exploration. The "three-act play" exercise requires inter-group cooperation and communication at a level rarely reached in high school classes. The group work is not just a technique to solve the problem of insufficient computers. It is integral to the inquiry planned into this course. In fact, when we started the course this school year, we had one computer for every student. We moved back to two and three students per machine when it became apparent that the students were missing the benefits of the interaction intended by the course materials.

3. A COMPLETE! Teacher's Manual. A novice can work from the teacher's manual. Every week, every lesson, has been planned with the utmost care. The manual is twice as long as the book. The manual covers how to copy diskettes, how to organize diskettes so that they can be reused for different lessons, how to arrange the students in effective groups, when to duplicate the supplementary materials for future lessons. It has answers to frequently asked questions, suggestions on promoting the course, duplicatable quizzes, detailed solutions to programming problems, lesson plans organized for the reality of the classroom. But it has more than the answers and guides to lessons. This Teacher's Manual communicates the spirit of the course as well as the substance. In this rapidly moving era of technology, teachers are hard-pressed to stay up with their students: this manual helps the teacher stay ahead.

The course is packaged in a most reasonable fashion. The teacher's manual and course diskettes are expensive, but the diskettes may be copied. We have operated some of our machines with Rascal from a Corvus hard drive. The student test is soft-bound, but can be reused by students. The sections on history of computers are excellent. The quizzes are tough, but focus on the main ideas of the language.

Computer educators have fought an uphill battle to broaden opportunities for students in computing. We have had to argue, plead, beg for funding. We have had to set up crash programs for training of teachers. The tasks have been difficult, even in this era of technological chic. The naysayers are already nibbling at the popular justifications of programming. How can we establish the validity of programming as an experience in broadening human thought and human power? Computer Power shows the way.

REFERENCES

Computer Power: A First Course in Using the Computer. Michael Moshell, Project Director. Gregg/McGraw-Hill, 1982.

Michael Moshell, "The Intelligence Amplifier: An Essay on the Design of Computer-Based Curricula," The Computing Teacher, April, June and September, 1982.

PASCAL AS A PROGRAMMING LANGUAGE FOR BEGINNERS

By: Robert Moll, University of Massachusetts, Amherst, MA

Pascal is by far the most popular programming language in American colleges and universities. According to some estimates, as many as 90% of all introductory college programming courses are given in Pascal. This is no accident. Pascal has a number of very attractive features. First, Pascal programs are easy to understand; instructions look almost like English sentences. And unlike many versions of BASIC, Pascal allows meaningful variable names. Pascal also outshines BASIC in the way it allows you to structure data. Pascal's most significant feature, however, is its procedure command, which encourages top-down design and "divide and conquer" problem solving. Programming with procedures teaches logical and systematic thinking.

In spite of its advantages, Pascal is seldom taught in American high schools. There are several reasons for this:

- Pascal has traditionally been a compiled, rather than an interpreted language.
- Pascal syntax is perceived as being formidable, and especially so in comparison to the very transparent syntax of BASIC.
- The Pascals that have been available for school computers have not been very easy to use.
- Most high school teachers do not know Pascal.

For a beginner, compiling a program is a tedious process. A student must first learn how to use a compiler and a linker, and this, by itself, can be a chore. Moreover, compilation slows a beginner down. The process of writing, compiling, running, and debugging a program and then compiling it again can be overwhelming for beginners. With a compiled Pascal, telling a student, "Here is a program. It does X. Change it so that it does Y. Then change it again so that it does Z," incurs considerable overhead.

Pascal syntax is another impediment for beginners. Pascal's semicolon rules can be hard to grasp at first, and the BEGIN-END pairs that seem ubiquitious in Pascal are an added burden for someone who is just learning how to program.

Are these impediments insurmountable? Can Pascal work as a language for beginners? For those who favor Pascal, there are several strategic reasons to hope so. First of all, compared to a student who knows only BASIC, a high school student who has already studied Pascal will be much better prepared for the computer curriculum at most colleges. Moreover, the College Board's decision to offer the advanced placement exam exclusively in Pascal makes the language a much more appealing alternative for schools. Because the AP exam requires three semesters of preparation, schools can save valuable teaching resources by presenting an all Pascal computer curriculum.

This brings me to two new Apple products: Macintosh Pascal, which Apple released in 1984; and Apple II Instant Pascal, which Apple will make available later this year. These two languages, which are almost identical, go a long

later this year. These two languages, which are almost identical, go a long way toward making Pascal a viable alternative to BASIC as a language for beginners. Here are some of the features that make these two languages so attractive:

- Both languages are interpreted. This means that programs are ready to run as soon as they are typed in. Programs do not go through a time-consuming compilation process each time they are modified.

- Both languages "pretty-print" program text as a program is typed in. This means that Pascal's formidable "BEGIN-END" structure, rather than being a drawback, now works as an aid to a beginner who can "see" the logical structure of a program as the program text appears on the screen.

- Both languages catch certain bugs as soon as a line of text is typed in. This unusual feature is the exact opposite of what happens with compiled systems, where even the smallest typo is not identified before compile time.

- Both languages provide an Instant window that allows you to execute individual instructions that are not included in a program.

- Both languages provide an Observe window for watching the values of variables and expressions during program execution. While both the Instant and the Observe windows are primarily debugging tools, they are also quite valuable for helping beginners understand program execution.

- Both languages make use of a substantial library of graphics commands. The graphics commands make it possible for even the simplest programs to produce exciting output. For example, looping can be taught using graphics commands that create animated pictures on the screen.

These features of Macintosh Pascal and Apple II Instant Pascal make them ideal vehicles for teaching beginners to program. To be sure, Pascal is not quite as simple as BASIC. But a semester of BASIC at the high school level does little to advance a student in a college's Pascal-based computer curriculum. Apple's two interpreted Pascal products are **almost** as easy to learn and use as BASIC; and they do a much better job of introducing students to the principles of programming.

WORD PROCESSING IS GOOD FOR ELEMENTARY TEACHERS AND STUDENTS

By: Karen L. Mueller, Butterfield School, Libertyville, IL

A mere year or two ago the content of most all educational computer meetings and workshops was centered around Computer Literacy and Programming. It was very important for all of us to teach our students the history of computers, all the vocabulary of the computer world, how to program the computer to do what we wanted, and the possible careers the students could have someday. It is now a new year and a new world of computer education seems to be on the horizon.

When I attend workshops I hear questions that ask how teachers can integrate the computer into their classes and into the curriculum itself and questions of what can the students do with the computer. The word APPLICATIONS comes to my mind when I hear these questions. One of the greatest computer applications for both the student and the teacher is word processing.

When I refer to a word processor, I am referring to a software program that allows the user to use their computer to manipulate the written word as it's entered on a computer keyboard. It is a tool that can greatly extend the written capabilities of children and adults.

There are many students and teachers who have trouble expressing themselves verbally. There are others who have poor penmanship and some who have trouble writing but can express themselves orally. Most of these people do not lack the ability to think. They just have trouble expressing themselves. Word processing will not solve all problems but it will help students and teachers express themselves better and with less frustration and more success.

The motto in many disciplines is "no pain no gain" or no hard work no learning. But that does not have to be the motto for using word processing. In word processing writers cramp is unheard of. The room is never cluttered with crumpled up paper with false starts. The reader does not suffer strained eyes from trying to read poor handwriting.

Poor organization of materials can be improved by using some of the features of word processing. A student should be encouraged to continue typing until he has finished his train of thought and then go back and edit any mistakes or reword unclear portions. The frustrations from having to write, rewrite, and write again no longer have to exist. Corrections can be made easily by inserting, deleting or moving text.

I have taught word processing to children in grades 3-8 and also to adults and find that they all love to use it. In fact children are willing to buy computer time to use the word processor if their teacher has such a plan. They love to write when they can use word processing. They also like to see the finished product. It does not make any difference how much or how little they write, the printer will always produce a paper that looks neat and to them that is great. Word processing can be taught to the gifted child and to the exceptional child with the same degree of success. Adults do not tend to get as excited. They can attend a workshop on word processing and say "that is nice", but they do not really believe that until they develop a need to use the word processor. Sometimes the magic comes from a college class that they are taking. Other times the need comes from the school newspaper that they

might help the students write. Whatever it is, once they start to use the word processor they will help spread the good news.

Spreading the good news is very important. Once teachers begin to use word processing themselves, the more likely they are to encourage the use of or even teach word processing to their students.

Teaching word processing is not difficult. The students need to know about seven basic items:

1. How to start-load-open a file.
2. How to insert text.
3. How to delete text.
4. How to use upper and lower case letters.
5. How to move the cursor.
6. How to end-save-close a file.
7. How to print.

Everything else is extra and can be learned as it is needed.

There are a few things that might help you teach word processing. Moving around the screen is important. Not all computers have four arrow keys and controlling the cursor with letters may take a little longer for students to learn. Try putting a maze on a transparency and tape it to the monitor and let them move the cursor through the maze.

Write a little paragraph using homonyms. Put both words in the text and let the students select the proper one and delete the word that is wrong. Write your own madlib and have the student insert words that could fit into the blank spaces in the paragraph. To save time from typing and spend more time on learning to edit, have students load in material that you have already typed. They can then spend valuable learning time correcting your intentional mistakes instead of typing the original.

Another way is to have them write a letter to their grandparents or to their parents. They will learn how to insert, delete and to use upper and lower case letters as the need arises. Print their letter often, stressing that it is only a rough draft. They should quickly appreciate the ability to correct a rough draft and have the computer do all the work of retyping the finished copy. Do not let them have the letters until they are in proper form. Remember to be a perfectionist. Making corrections and changes is easy and it is very nice to see a finished product that the student can be proud of. For some it will be the first time that they have had a neat end product. Their joy will be your joy.

I have used four word processing programs and I think that they all are good. BANK STREET WRITER was the first of the easy-to-use low cost word processing programs. My next love was HOMEWORD, as it is so easy-to-use and has many of the features of the expensive word processors. The most recent is MAGIC SLATE. The large print that this program produces will delight primary teachers. This program, with many features, promises to be a good program for children and adults.

Whatever program you have, it is most important that you use it and let others know how pleased you are with word processing. Encourage students to turn in work done on a word processor. Help them to learn how they too can improve their writing. You will be very pleased with the results.

KEYBOARDING

By: Mark Mueller, Highland School, Libertyville, IL

Many people of the world are very concerned about the population explosion. They wonder if the earth will have the resources and if mankind can produce the food that will be needed by an ever increasing number of people. Many educators are also concerned about a population explosion. Rather than the increasing number of people, however, they are concerned about the ever increasing number of microcomputers. They wonder if people will be prepared to use them effectively.

Because computers are popping up in all areas of our society the ability to keyboard is fast becoming a required basic skill. Many schools have computers and are buying more. They are trying to prepare students for the world of computers tomorrow. Mastery of the keyboard will enable a student to concentrate on the task at hand rather than search for the proper keys. Not only will he have an advantage as a student, but he will also have a definite edge when he leaves school.

To help students prepare themselves for their future, many schools are putting a keyboarding program into the curriculum. Keyboarding classes are different from typing classes in that a keyboarding class would be concerned only with putting data into a machine. The typing class would additionally be concerned with the correct arrangement of letters, tables, and reports. Care needs to be taken prior to setting up a keyboarding program. If it is not done right the students will gain very little.

It must be remembered that keyboarding is a skill that needs frequent practice to master. It can best be learned through a program of organized instruction. It ideally should be done at all grade levels, beginning early in a student's educational experience.

Much thought went into the development of the keyboarding program at Highland School. Although it may be desirable to start children earlier, because of the placement of computers in the district it was decided to begin the keyboarding program in sixth grade. As students don't have an extensive computer course until seventh and eighth grades, this decision seems to fit in very well with our curriculum. Students take keyboarding during their arts block period. Everyday for six weeks they will have keyboarding. At the end of the six weeks they will rotate to another arts block course such as music, art, or shop. Keyboarding is taught in a lab of networked TRS-80 model III computers. Each student has his own computer for 40 minutes.

As the future teacher of this keyboarding class, I was very concerned that we find some good quality software that will help teach keyboarding. I could type fairly well but my only instruction in keyboard education was the typing class I had had in the seventh grade. After looking at several programs we decided on the Alphabetic Keyboarding program from South-Western.

South-Western seemed to have most of the characteristics needed for a good keyboarding course. Based on what reading I had done, I was looking for a program that would meet several criteria. First, and probably most important, it had to teach correct techniques. By practicing correct techniques a student should continue to improve both his speed and his accuracy.

Many students already have computer experience. Some are even very proficient at a hunt and peck method of keyboarding. This style is often reinforced by computer programs that require one letter or short word responses. When it comes to typing sentences and paragraphs, this style can be very slow and frustrating for students. It was important, in my search for software, that I find a program that had practice typing sentences and paragraphs and not just letters and words. It is also important for the words that are practiced to be real words and not just nonsense letters.

I have generally been pleased with the South-Western program. Although the program prevents incorrect keys from appearing on the screen during practice, when students are being timed there is no checking for mistakes. I was really surprised at the number of my students who could type over 350 words per minute. They were upset when I told them that it didn't count if they had mistakes in their sentences.

Using CAI to teach keyboarding is not as easy as I had first anticipated. The students are very concerned about speed. Many see others hunting and pecking who can type faster than they can. It is a continuous struggle to make sure that the students are practicing the correct techniques. Many rush through the lessons without bothering to read about proper finger placement. I must continually watch their fingers and constantly remind them to keep the home-row covered and to reach with the proper finger. Those who do work on the correct techniques will soon overtake the students who are still hurting and pecking.

Students begin keyboarding class with a lot of excitement. They soon become tired, however, and will complain of being bored. I have found that doing something different every few days can help. I may give them a worksheet or test or even let them run some of the other software in the computer lab. I have found that the stereo in my room can be the best defense against boredom. Students will take turns picking a station to listen to or may bring records from home. Working at the keyboard with their lessons goes easier with the music to listen to.

My most difficult task with the keyboard classes has been trying to find a good way to evaluate the student's progress. I would like to concentrate at this level on the techniques developed by the student. I find it very hard, however, to watch a student's posture and fingers and then try to say whether he is better or worse than another. After every few lessons I will give them a quiz on the keyboard. I may ask them to fill in a blank keyboard with the letters that they have learned so far. Or I might have them trace their hand on a sheet of paper and then write inside the fingers the letters that that particular finger will press. I have had the most success with a short program that I wrote in BASIC that gives the student four sentences to type. Each sentence is timed and when finished the students time and his sentence are printed on the printer. I can then count the mistakes in each sentence and compare that with other students and their times and scores. To try and catch those students who may still be hunting and pecking, I have the students type the sentences again, but this time with a towel over their hands so that they cannot see the keys.

I am anxious to have these students again next year as seventh graders when they have a six week course on computer literacy. At that time I should be able to tell if their work at keyboarding during sixth grade was of any benefit to them. Hopefully they will retain what they learned and be efficient at the keyboard.

Keyboarding is an important skill that today's students should master. If you are interested in beginning a keyboarding program, there are many different ways to go about it. It is critical to remember though, that keyboarding is a skill that takes lots of practice to master. Care should be taken that students learn the proper techniques and practice them. They also should be practicing on sentences and paragraphs rather than entirely on short words. With the proper instruction and supervision, these students should have an edge when they begin to use those computers out in the real world.

TEACHERS TALK—THE FAILURE OF COMPUTERS IN EDUCATION?

By: Thomas C. O'Brien, Southern Illinois University, Edwardsville, IL

In the early 1970's a school principal told me that The Open Classroom didn't work. "We tried it," he said. "We moved all the desks and chairs from the classrooms into the halls. And the kids sat on the bare floors of the classrooms. It didn't work. The kids got cold bottoms. We kept it up for two weeks, but then we moved all the furniture back into the classrooms. The Open Classroom was a bad idea."

Sadly, much of the history of American education is a history of bandwagons, ideas which were adopted rather than adapted, which were installed without much serious thought about purposes and underpinnings, and which were parachuted from on high without teachers and administrators engaging issues in a thoughtful and serious way. The bandwagons came in a flurry, promised sun, moon, and stars, and vanished without a trace. Some of them deserved to vanish; witness the absurd interpretation given Open Education above.

What's the fastest vehicle in a vacuum? A bandwagon!

Now we have another bandwagon on our hands -- computer assisted learning. It will not be long, I predict, before American education throws up its hands and cries, "We tried it. We gave it our best. Computers don't belong in the classroom!"

But are computers being given a try?

I am an author of computer software. I am a user of computers to the extent that I can reasonably say that they have transformed my life. (Virtually all my print material, professional and private, goes into a computer and nightly I check into my electronic mailbox.) I am a teacher educator and a researcher on the intellectual development of children who has written widely about computers-in-education (See Classroom Computer Learning, September and October 1983, and Phi Delta Kappan, October 1983). It is clear to me, and to many others, that computers are spectacularly versatile tools which increase one's personal firepower almost without limits.

There is no question but that computers can play a major role in education. That they will is far from certain.

Recently I conducted a course on computers-in-education. Some 25 elementary school teachers took part. During the course, I asked participants to describe the state of the art as they knew it locally. Then I asked, what would they want to happen?

What's happening in local schools? Here are the comments, paraphrased for style. I have separated the comments into three sections: hardware, software, and in-service education. What would teachers like to happen? That section follows. Then comes the author's summary and commentary.

HARDWARE

"The principal buys the hardware and sticks us with it as though to say, 'I've done my part. You're on your own'."

"My school administration bought a boatload of TI computers just after TI left

the education market. Why? They were on sale at $49.95. The problem is that there's virtually no educational software for the TI so we've shelved them."

"My principal bought a bunch of cassette driven Pets because they were cheap. We've got a bit of software, not much, but now he's ordered some Pets with disk drives but no new software. We're having one of our seventh graders translate from cassette to disk over the summer."

"My school has no computers at all."

"My school has 4 computers for 3500 students in K-12."

"The PTO wanted to support the purchase of computers but no one knew what to do, so we settled on price as the criterion. The computers are on the shelf because there's no software."

"We have one computer in our school. It's upstairs. It never gets downstairs."

"Every school in my 20-school district has one computer bought with district funds. They are used solely for Instructional Management System record keeping. There are some 200 other computers in the district, but these were all purchased with PTO funds."

"A custodial mentality seems to prevail. Buy the machines (at the lowest cost possible and without much attention to their educational use), then protect them. Don't, for Heaven's sake, let kids touch them; they'll break them!"

"Computers in my school are used for administrative purposes only -- record keeping, principal's letters, etc."

"A lot of school administrators say that they're gung-ho about computers in education. But their heart isn't in it and they often don't know what they're doing. They want to look up-to-date to the community."

SOFTWARE

Teachers in my school thought that the computer did it all. They didn't know that there was such a thing as software. Buy the computer, they thought, and you're on your way."

"I am amazed that there's other software than MECC software. We were told that MECC was the only software producer."

"My school district has a policy that drill and practice software is the only stuff the slow kids get to see. I may be wrong, but I think that all kids -- especially these kids -- need to see something other than rote connections. If they don't see something different, they'll continue to sink in the sea of rote meaningless memorization."

"I am proud of my colleagues. The only software they knew of was provided by one company, and it was dreadful. Many of them refused to use a computer in their classroom because their professional conscience was violated."

"My school uses computers but I don't. They only use of computers in my school is that something like 37.14 X 778.23 is shown on the screen, the kids work the answer out by hand, then the teacher pushes a button and kids check their answers. That's not good education."

IN-SERVICE EDUCATION

"We've had no in-service education in our district. There's a once-a-year county institute where some of the sessions deal with computers in education."

"We had a half-day workshop -- 3 computers over 100 teachers -- at the beginning of the school year. It took place on the afternoon of the first day of school, just after we had met our new kids for the first time. We had gobs of decisions to make on the basis of our morning's meeting with our new classes and no one paid much attention to computers. Nothing has taken place since then."

"We are expected to squeeze in a knowledge of computers amidst everything else. No time is provided to give the issue any real attention."

"A bunch of us took a course on computers. The course amounted to elementary computer programming. That wasn't what we were looking for and we dropped out in droves. Those who finished the course were able to produce rudimentary programs which drilled kids on state capitals, exactly the sort of thing that's in workbooks. Workbooks or computer software, I ask whether state capitals are important issues for kids to engage."

"Most of my colleagues have no idea that computers can be used for word processing, for graphics, for music, for problem solving, for old and new topics and new approaches to math and history and language, for information services such as CompuServe and The Source. The computer is a drill and practice machine, they think. They don't know that software publishers such as Conduit, Spinnaker, Sunburst, the Learning Company, Scholastic, MECC, TIES, and CCW even exist."

"We took a LOGO course and we spent all our time drawing pictures. There was no attention given to the power of LOGO as a computer language, to the important issues of LOGO (such as recursion, variables, nesting of procedures, etc., which are immensely powerful intellectual tools), or to the non-graphics uses of LOGO."

"Our school has a Computer Priestess. She took a week's workshop and that put her far above the rest of us. She only works with games and elementary programming. She's reluctant to help us, to share her priesthood. She won't share her manuals or her software. If we want help, she'll do the work rather than help us to do our own."

"Some of my colleagues are using a computer just to be able to say, `I'm using a computer.' They have no real feeling about how a computer might fit productively into their classroom."

"The way we were introduced to computers was wholly inadequate in terms of time and depth and gaining useful knowledge. What would you think of a NASA astronaut training program which consisted wholly of a half-day's rush through mission Control in Houston? You'd scream about your wasted tax moneys."

WHAT WOULD YOU LIKE TO SEE?

"School administrators should think about computers as new and important educational tools, not as hardware-like desks and urinals. They should focus on the role of computers and software in education, not on mere dollar savings and not on computers as a way of gaining good PR for themselves."

"School administrators should spend the time and the energy and the thoughtful reflection, in collaboration with teachers, that computers-in-education deserve and demand. If not, they should turn over the reins (and glue support) to those of us who know (and who want to know more) about the issue."

"Teachers should have time and support to make the issue their own. They should have several months of constant access (perhaps in their home as well as in school) to computers and gobs of good software before using computers in the classroom. One hundred hours of interaction with good computer software should be a prerequisite for any teacher planning to use a computer with kids. That given, teachers should have constant opportunities to acquaint themselves with new materials. This is a fast track. A one-time introduction, however lengthy, won't do."

"Beware of salesmanship and slick marketing and glossy packaging. Beware of colorful screens masking poor education. We're dealing with the lives of kids. We're not selling aspirin or Buicks."

"Beware of company kickbacks to assure purchase of hardware and software. 'You're a heads-up educator. Can I buy you a dinner at the Four Seasons?'"

"Teachers should have easy access to print background on computers-in-education and they should have easy access to hardware and software reviews. But don't swallow someone else's review whole-hog. The best test is one's own tryout. And don't discount word-of-mouth (grapevine) comments, which often have more veracity than the professionally produced review."

"Most of the good educational software comes from small publishers. Let's dig and find out what's available and not settle for something just because a salesman brings it to your front door. About half the software publishers, I've learned, have a 30 day free preview policy. Let's tell people that this is so and let's encourage and share serious evaluations."

"School districts should provide roving educational and technical consultants, something like the educational advisors in the British schools. Educational advisors -- perhaps teachers on a rotating basis -- would spread ideas. Technical advisors would fix small glitches (perhaps even by telephone advice), thus saving the 30 days' down time often involved in off-site repairs."

"Let's move away from priests and priestesses and intimidating technical language and a focus on electronics. It would be laughable to get so technical with other commonplace tools, such as telephones or televisions. Let's use this tool for the well-being of children. Let's marshal the power of computers in education and stop diddling around with self-serving esoterica."

"Don't ever feel that you're finished. This computer stuff is constantly growing and it helps you to grow. I have the feeling that `the sky's the limit.'"

CONCLUSIONS AND COMMENTARY

The comments here came from elementary school teachers in Silicon Desert. Whether they are generalizable over the nation, I don't know, but I suspect readers will recognize some of the points as pretty close-to-home.

What springs out, I think, are several points:

1. Teachers are honestly concerned about the well-being of children, their judgment is often on target, and they are often ill-served and ill-supported in their professional quests.

2. Computers-in-education, one of the only educational movements in concert with American societal trends (unlike "modern math" and ITA and dozens of other bandwagons over the past twenty years) offers great promise to American education. The reports of teachers suggest that the history of previous bandwagons is being repeated. Computers are not being seriously engaged as educational resources.

3. On one hand, teachers are treated as pawns with ideas and policies parachuted from above. On the other, they are invited to innovate, so long as they do so in their spare time, with only nominal support, amidst the burdens of ongoing obligations. Can't teachers be viewed as essential partners in educational advancement? Can't time and support be provided them to do a worthwhile job?

4. We are "tacking on" computers to an already overcrowded and disarticulated curriculum.

For example, we spend endless hours teaching arithmetic computation to elementary school children with exhaustive (and exhausting) attention to speed and accuracy at a time when 4 million long calculations can be performed in one second. Could we engage children's time and intellect more fruitfully? Need we continue to prepare children for a world which is long last?

Perhaps we should re-think the curriculum in terms of the availability of computers in the home, schools, and industry now and in the Twenty First Century, where present children will spend most of their lives. Computers are super at memory tasks. They are not very good at thinking. People are very good at thinking but not so good at memorizing. Have we got things the wrong way around in the present curriculum?

Does our present curriculum prepare children for the lives they will lead in the Twenty First Century?

5. Will we hear, two or three years from now, "We tried computers in the classroom. It didn't work."? Local evidence suggests a clear cut. Yes, unless educational improvement is undertaken as a fundamental task by teachers and school administrators.

THREE ANCHORS AND COMPUTER-ASSISTED LEARNING

By: Thomas C. O'Brien, Southern Illinois University, Edwardsville, IL

"Anchor" means, among other things: 1) a heavy object, often with hooks, connected by cables to a ship; 2) that which gives stability and security; 3) the last runner in a relay race.

I have another meaning. By "anchor" I mean that which keeps us rooted to the same place, which prevents progress, which forestalls change. There is a lovely poster I've seen -- a sailing ship running before the wind with the caption "Ships were not made for safe harbors." Ships need anchors at times, but not at ALL time.

Three educational anchors which limit children's intellectual sailing are: 1) the anchor of print, 2) the anchor of facts, and 3) the anchor of transmission. In each case, the availability of microcomputers and good software can enable us to cast away these anchors, these heavy hooked objects which are so deeply submersed that we sometimes forget they are there.

THE ANCHOR OF PRINT

Much of American education is dominated by print. As early as first grade, the curriculum IS the textbook. Math IS the writing of numbers and equations. Science IS the filling out of workbook pages. Language and social studies ARE words in a textbook and the circling of multiple choice items.

The major role of education is the development of intelligence. And the very nature of intelligence is action -- action of body and mind. Today's school curriculum often precludes action, especially that important action called thinking. Print too often becomes an end in itself and we all too often lose sight of the important ends that print can serve on behalf of intellectual action.

One of the most important intelligent actions, classifying, has its foundations in kids' putting things into cartons. Number, one of the most important ways of making sense of and describing our environmental soup, begins with touching and pointing and counting, not with the scribbling of things in standard form on a slate or a paper. Language has communication -- the interchange of information and ideas and argument -- as its major function. In today's schools, language is often stuff that is written on paper and graded by teachers. And science is the search for regularities and relationships among real world phenomena, not the memorization of words in a list or a diagram.

The American Indian, I've been told, loath putting things on paper. If one writes about something, the thing loses its spirit. Many ideas lose their spirit in American classrooms when everything comes, goes, and is graded in a paper-and-pencil format.

Don't get me wrong. Print is a major intellectual construct. It is a means of communicating and it is a means of recording the actions, the thoughts, and the creations that one has made. In an instant, thanks to print, I can be in Sixteenth Century Rome or in the mid-Atlantic with Columbus or in the first days of Lincoln's election campaign. Print has its place and a very important place it is. Lincoln's election campaign is a situation which, by its very

nature, one must experience second-hand. But must all school experiences be second-hand? For example, must equations such as 2 + 3 = ___ take the place of actual experience and action with twos and threes of real-life objects?

A friend of mine recently posed the following problem to his class of middle class suburban fifth and sixth grade children: $5\overline{)15}$. Of course all kids wrote a 3 above the line. Then he dumped a pile of poker chips out and asked each of them to show what they had done. Of twenty-two children, only two drew out 15 chips and separated them equally into 5 (or 3) piles. Division for 91 per cent of the children was a mechanical procedure acted out in print, with nothing to do with things or actions in the real world.

Much has been written elsewhere about educational software which conveys the same static stuff and inert procedures as textbooks. The words "electronic workbook" have been widely and aptly used as publishers rush to the software marketplace. Damn such software or ignore it at risk. Its contribution, in the long run, is major. Teachers are finding out how boring the stuff is. It's hard for children to turn off a textbook, but it's clear-cut when a kid turns off a computer. Children, teachers are finding, are voting with their on-off switch.

The fact is that there's a great deal of software that's light years away from "electronic workbooks." This software enables ideas to keep their spirit by causing children to act, to think, to create. The interactive challenge such software provides to young (and old) minds amounts to no less than an educational revolution, allowing educators to do old things better and to do new things never before dreamed of.

There are four aspects to the situation:

1. School children should have first-hand experience with real objects and actions and events rather than learning about them second-hand through static print (whether on paper or on screen). But some situations are out of reach for reasons of time, cost, or efficiency. Here computer simulations can play a great role.

The tossing of dice is an important aspect in learning probability and combinatorics. For example, few teachers could stand thirty kids' tossing two dice enough times to build up a decent histogram. The solution: use a dice-throwing program. A thousand tosses can be made in a matter of minutes, complete with both actual and theoretical histograms. The creation of simpler dice simulations are within reach of the average eighth or ninth grade programmer.

Capture an event from long ago? Sure, read about it or watch a film. Take part in such an event? Sure! Such a thing is possible through simulations such as Oregon Trail and Civil War.

Want to study physics in real life instead of memorize formulas? Try Scatter (Chelsea Science Simulations), where (without vast equipment expense) one can experiment with atomic particle scattering.

2. There are gobs of software becoming available in which children can interact with ideas and relationships. This software simulates real life as well but its purpose is not merely the re-enactment of events. Software of

this kind deals with the interactive construction of thought processes.

Take a look at Counters, for example, where beginning notions of number -- 1-1 correspondence up through beginning addition and subtraction -- are tackled with an emphasis on the actions and ideas involved, not on the printed, static record of those actions.

Look at Gertrude's Puzzles where children act on logic blocks and construct important ideas about classification and inference.

Or try Baffles where one generates and tests hypotheses and learns a great deal of if-then thinking, an issue notably impervious to transmission via print.

3. Earlier I railed against the tyranny of inappropriate print. But print is a vast and welcome ally as a means of conveying ideas. Here too, computers and good software play a major role. Computers can lighten the burden of children's print production.

Imagine a child's having produced an imaginative story about Cinderella or imagine a group of students having produced a story in the round about Columbus. Some thirty or forty-five minutes may have gone into manuscript production. With their teacher's help, they've edited the story and set about correcting misspellings and grammar. With manual rewriting, they're in for another 30 or 45 minutes of writing, much of which is sheer drudgery. With word processing software, simple enough for use as early as grade 1 (perhaps they can make corrections with a keystroke or two. The March 1984 issue of Electronic Learning lists more than 30 different word processing programs currently available for personal computers.)

Further, with graphics utilities newly available -- Koala Pad, Mouse Paint, and Chalkboard among others -- it's quite realistic to expect that children will routinely begin to write, edit, and design books for their friends, for younger classes, and for the school library. Quite a richer task than plowing through workbooks, electronic or otherwise!

SUMMARY

Computers and idea-oriented computer software can help to overcome the tyranny of inappropriate print and maximize the effect of productive print. Print has a valuable, indeed essential, role in education, but first-hand action and experience are at the heart of learning. Educational software can help to provide action and experience in a rich variety of ways. Knowledge is not a spectator sport.

THE ANCHOR OF FACTS

"They must learn their facts," teachers tell me. "They must memorize that 2 + 3 = 5."

What facts need to be known by Everyone is not a trivial question.

The very word "fact" comes from the Latin "facere" -- to do, make, act. Yet we in education often regard facts as things, not anything dealing with human doing or making or acting. Worse, we regard facts as immutable. (It was once a fact that malaria was caused by bad air, hence its name. What other "facts" have fallen by the wayside in your lifetime?)

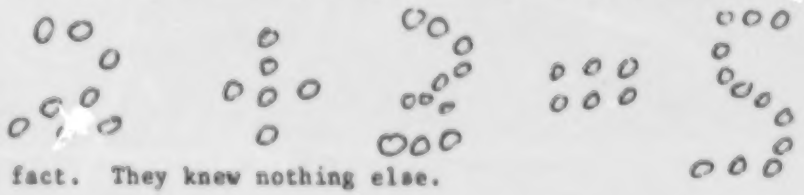
Worst of all, many teachers regard "facts" as the first necessary steps in knowledge. "Get those static things established first," they seem to say, "and only then go on to higher things such as thinking."

But look at the first steps in knowledge. Look at the extreme case, the beginnings of knowledge in infancy. Newborns construct, with vast (and universally replicated) complexity the notion of nutrition. They suck anything that brushes their cheeks. They slowly and gropingly construct the notion that a nipple is fruitful, that a pillowcase is not, that a thumb is a pretty rewarding substitute.

Are facts things? No, they are constructs of human mind. Are they immutable? No, take a look at the history of science. Are static facts necessary as building blocks before any thinking can take place? No, even the newborn infant is thinking -- generalizing and discriminating -- from the very start. What a wrong view of knowledge we give our kids in school!

Recently we had the opportunity to work with teachers in looking in on "facts" as known by children. We gave children a bunch of chips (penny-sized) and asked them to show that 2 + 3 was 5. Many children used the chips as follows:

art 2 + 3 = 5



Yes, the kids knew a fact. They knew nothing else.

The same spirit that we educators endow to facts -- God-given, immutable, necessary building blocks before thinking can take place -- is given as well to procedures. Kids in school spend endless hours practicing procedures. Kids in school spend endless hours practicing procedures handed over to them full blown from textbooks through teachers.

Art showing facts and procedures

We recently had an opportunity to look in on children's knowledge of procedures. Based on the informal work of a local teacher, Shirley Casey and I posed five multiplication exercises for children in grades 4, 5 and 6. The kids performed quite well -- 75 per cent or better on questions such as 13 x 12 = ?. One of the exercises was 6 x 3 = ?, a question answered correctly by 97 per cent of those questioned. We then asked the children to produce a word problem or a real life situation for 6 X 3 = 18. At grade 4, 75 per cent of the children could not do so (although repeated addition, such as "Monday I bought a six-pack of Pepsi, Tuesday I bought a six-pack of Pepsi, etc." would have been scored as correct). At grade 5 the figure was 85 per cent! And at grade 6, 30 per cent were still unable to come up with a multiplication story for 6 X 3 = 18.

As interesting and shocking, at least half of all erroneous responses at each grade level involved straight addition: "Six boys came to a party. Three girls came to a party. How many people were at the party? 6 X 3 = 18. That's the answer." Teachers with whom we work questioned the results. "Must

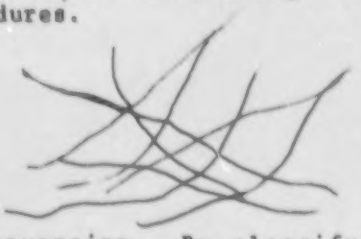
have been a bad day," they said. "A bad group." "Bad teaching." "Try it with your kids," we said. The result: virtually identical. One teacher just reported that 50 per cent of eighth graders (!) in her school district failed the task.

It should be clear that our present emphasis on static facts and procedures leaves something to be desired.

Perhaps the most important fact about facts is that the proliferation and rapid dissemination of information in our world had made absurd the notion that education should consist of the transmission of facts and procedures. One's head is simply not large enough!

Children -- we all, in fact -- need to learn to ACT on information. What does this mean? It means that we should help children to locate, to make sense out of, and to use information (rather than merely store it) and we should encourage them to generate new information. Chiefly, it means that children in school should construct relationships rather than merely warehouse facts. It means, for example, that 2 + 3 = 5 because 2 + 2 = 4 and 2 + 3 is 1 more than 2 + 2. It means that children construct fabrics of knowledge rather than individual strings involving facts and procedures.

art showing fabric



How does one act on information? By sequencing. By classifying. By inferring. By asking, "What if...?" By logically multiplying. By discerning between relevant and irrelevant. By distinguishing between possible and impossible, between plausible and necessary, between necessary and sufficient. These are some of the intellectual acts that one constructs during the school years. At present these issues are pretty much ignored by the curriculum because there are too many facts and procedures to attend to.

Fortunately there is an increasing amount of computer software which enables children to act rather than merely to store, in which knowledge is constructed rather than copied, in which coherence and meaning and generalizability rather than accumulation is the goal, and in which fabric rather than threads is the payoff.

There are at least four arenas which fit these conditions:

1) Information and communications services, such as The Source and CompuServe, which enable students to satisfy their curiosity through instant contact with vast amounts of up-to-date information (and with one another or with experts in a field).

2) Simulations, such as Bird Breed, which enable students to investigate the course of biological, physical, chemical, ecological, mathematical and economic phenomena. Such simulations -- see Population Dynamics, for example -- enable interactive investigation. ("Let's see what the effect of such-and-such variable might be.") They are not static linear sequences.

3) Graphics and language aids, such as Delta Drawing, Koala Pad, and Bank Street Writer, among others. One major aspect of intellect is representation (from the Latin res praesantans, making a thing present), an issue widely ignored in present day schools.

4) But the most important aspect of educational software is its role in fostering students' thinking.

Human knowledge does not grow from static facts and procedures. Rather, it grows from one's interactions with situation (with mistakes and leaps and discontinuities) and the construction of relations and patterns and logical necessities thereby.

Take a look at Baffles (hypotheses testing), or Gertrude's Puzzles and Rocky's Boots (logic), or The Factory (combinatorics, geometry) and The King's Rule (plausible necessity), or Taxman (algorithmic thinking), among many others now available to schools and homes.

SUMMARY

Knowledge is not static stuff. Facts are the result of doing and making and acting. Computers can enable children to do and make and act. Let's not settle for mere warehousing. Knowledge is not a spectator sport!

THE ANCHOR OF TRANSMISSION

Walk around your school -- or any school from elementary school through graduate school. Look in on classrooms. What you'll see, all too often, is the transmission of information. There'll be one person who's active -- the teacher -- with the students passively listening or taking notes. Or the students will be working solo at their texts and workbooks, copying or practicing facts, conventions and procedures handed over full-blown to them. It's rare, I'll bet, that you'll see significant intellectual activity on the part of students.

The origin of the word "student" is the Latin studere, to be eager about. My impression is that American school children are eager up until about Grade 4. After that they are numbed by the passive role thrust upon them. All too often, children are b-o-r-e-d.

Knowledge is not "stuff" to be handed over like a lump of coal, nor are children empty vessels capable only of storing information. Knowledge is a fabric of ideas, meanings, relationships, inferences, intuitions and quests which one constructs in interaction with outside reality. Knowledge has two complementary aspects -- it tends toward coherence, stability, economy and generalizability at the same time that it seeks new instability, new novelties, new dissonances to resolve. In short, knowledge is organic. It develops and at the same time it sends out searches for new development. It's like a plant rather than a lump of coal.

No wonder school children are b-o-r-e-d. Transmission renders their minds storage bins.

Central to the construction of knowledge (i.e. learning) is the notion of interaction. You act on the world and the world acts on you. For example,

you are on vacation and you drive through a rural village. It reminds you of your childhood home town. Your view of the rural village is colored by your existing "home town" fabric and your original fabric adjusts itself to admit the rural village as a related entity.

That is to say we do not store reality in its raw form. We screen it, interpret it, transform it, even deform it in terms of our existing fabric. And in the interaction our fabrics evolve to accommodate the new reality.

Central to the learning situation is the notion of mismatch. Given a moderate amount of mismatch between the situation and our present fabric, we tend to reassemble the fabric to adjust to the new information. Too much mismatch and we turn attention away. Too little and we are bored. Thus mismatch is THE major aspect of the learning situation.

All this means that knowledge and learning are personal, not only in terms of "rate of learning" (a rather trivial variable) but in terms of the very nature of what is learned. What's a mismatch for one person may be an overmatch for a second person and an undermatch for a third. And, even if one mismatch is appropriate for several different people, it's virtually certain that their resolution of the mismatch will be different because of its interaction with their individual fabrics.

Thus moderate mismatch through transmission to thirty people in a classroom is usually impossible, unless the issue is trivial. In short, transmission, however efficient it looks on the surface, is terribly inefficient. We cannot expect mechanistic methods to enrich organic minds.

So here comes the microcomputer into education. "The intimate machine," a phrase coined by Sherry Turkle (Science 84, April), the computer can confront the individual learner with just the right amount of mismatch and thus promote the construction of ideas and relationships.

In immediate mode -- that is, without software -- one can tutor the machine to do just what one wants it to do. "Print out the first hundred even numbers" and "Write my name exactly one hundred times" are issues which can engage children as early as first grade. What an ally in the growth of knowledge and what a willing helper in carrying out the drudgery of repetitious operations!

It used to be the case -- for some of us it's only a generation ago -- that one's farthest reach into the outside world was that provided by the annual village fair. Now the personal computer with a modem can tap a whole world of news and information and games and personal interaction at a cost of pennies per hour. What to choose, among the vast variety of interaction available? It's up to you, the user, to tap into what interests you, what satisfies you, what itches your curiosity.

The Piagetian researcher, Mimi Sinclair, said it succinctly in her work with local teachers at the Teachers' Center Project, "We should see children as `under construction'. Even better, we should see children as `under construction -- self employed.'" No wonder school children are b-o-r-e-d when all the construction is done for them and their only intellectual action is to memorize.

But it's educational software that concerns us here as educators.

The "intimate machine" coupled with good software can provide mismatch for a wide variety of learners in schools and homes.

In Baffles, for example, the learner can start with five baffles, and issue accessible to most fifth graders, or with fifteen baffles, an immensely complex problem for most adults. As important, it provides the freedom for one's own construction of ideas -- mistakes, hunches, refinement of tactics, logical certainty, and construction of fabric. Perhaps most important, Baffles deals with issues of importance: the construction of inferential thinking.

I don't mean to cite Baffles in particular. I do mean to use Baffles as an exemplar of good software. It provokes important growth through the student's own efforts and it makes available a high ceiling and a wide variety of entry points.

I once asked the great problem solver, George Polya, what the characteristics of a good problem might be. "First of all," he said, "the problem should be interesting. What's interesting for one person might not be interesting for another. Second, a problem should be difficult, else it is not a problem. What's a problem for one person may be a trivial exercise for another. And most important, a problem should `go somewhere' and that `somewhere' should be important. There are many issues which are interesting to children, and many of them are difficult" (as in building the world's highest sand castle) "but the issue may not `go somewhere' of importance."

It is not out of order to bring Polya's advice to educational software. Let's encourage the production of software which is interesting. That software should have a wide variety of entry points and a high ceiling. (That is, it should provide the right amount of mismatch.) Best, the software should encourage important constructions -- constructions which "go somewhere" on behalf of the child and on behalf of the society and the world in which the child lives.

Let's not be satisfied with software that transmits static stuff. Let's not build rude huts for children when in fact children are capable of constructing cathedrals.

SUMMARY

In brief, the computer can bring life to b-o-r-e-d children and boring classrooms. It encourages what is missing in many classrooms but everywhere in everyday life -- the interactive construction of ideas. Knowledge is not, after all, a spectator sport.

INCREASING FEMALE INVOLVEMENT IN THE USE OF COMPUTERS

By Julie A. Poage, Roycemore School, Evanston, IL

The acceptance of the computer into education has presented numerous problems, of which the under-representation of females has become apparent (Anderson, Welch, & Harris, 1984: Center on Evaluation, Development and Research [CEDR], 1983; Kreinberg & Stage as cited in Bakon, 1983; Lockheed & Frakt, 1984; Revelle, Honey, Amsel, Schauble, & Levine, 1984; Sanders, 1984; Van Gelder, 1985.) Why are fewer females actively involved in computer use?

Females in the elementary grades participate in computer use in a fairly equal ratio to male students, however, the enrollment of females in programming classes falls as their age level increases (Alvardo, 1984.) In addition, females tend not to spend free time on computer activities.

Why are females less likely to use computers? One possible answer is the association many schools have made between computer use and math. Many studies have shown that math anxiety may translate to computer anxiety when computers are housed in math departments or taught by math teachers (Sanders, 1984.)

Limited computer resources in many schools places great competition between students for computer time. Females tend to yield to more aggressive and confident male students, thus, increasing female anxiety toward computer use.

Both educational and recreational software production has been aimed at a male population. Almost all current adventure games are developed around male fantasies. While many successful educational games have been fashioned after popular arcade games, with exploding rockets or space invaders.

How might educators encourage greater female involvement in computer use? Counselors should point out the numerous career opportunities available to students with programming experience. Computer courses should be designed to include appropriate female subject matter (i.e. textbooks displaying numerous male-oriented problems, such as baseball averages, should be avoided.)

Reduce competition for computer use. If computers are banked in a computer lab, allowing a "girls only" time might encourage greater female use. Scheduling required class activities or assignments involving the computer will bring more females into computer contact. If computer activities become a part of classes outside of math, students experiencing math anxiety may not translate these feelings to computer use. Encourage computer participation in art, English, social studies, science, and music. The computer need not be simply a math tool.

Share computer ideas with fellow teachers. This will lead to more female educators using the computer with confidence presenting a positive role model, just as many (male) math teachers have done in the past.

Educators responsible for the selection of computer software should be aware of possible sex biases in that software. Are females represented in leadership roles in the software? Are subjects presented that appeal to both males and females? Does the software concentrate on competitive, aggressive arcade games that might appeal more to males than to females? Are female

fantasies ever presented, as male fantasies are in adventure stories?

The equal involvement of females in computer use will become crucial as technology enters the classroom. If females are not encouraged now to take an active part in computer use, an extreme equity problem will result, not only in the schools, but also in the business community. Computer educators must make wise choices structuring programming courses, selecting nonsex-biased software, and encouraging fellow teachers to use computers regularly in the arts and social sciences.

REFERENCES

Alvardo, A. (1984). Computer education for ALL students. The Computing Teacher, 11(8), 14-15.

Anderson, R., Welch, W., & Harris, L. (1984). Inequities in opportunities for computer literacy. The Computing Teacher, 11(8), 10-12.

Bakon, C., Nielson, A. & McKenzie, J. (1983). Computer Fear. Educational Leadership, 41, 27.

Center on Evaluation, Development and Research. (1983, May). Sex differences in computer training: Why do boys outnumber girls? Phi Delta Kappa (Research Bulletin).

Lockheed, M., & Frakt, S. (1984). Sex Equity: Increasing girls' use of computers. The Computing Teacher, 11(8), 16-18.

Revelle, G., Honey, M., Amsel, E., Schauble, L., & Levine, G. (1984). Sex differences in the use of computers. Paper presented at the American Educational Research Association Annual Meeting, New Orleans.

Sanders, J. (1984). The computer: male, female or androgynous? The Computing Teacher, 11(8), 31-34.

Van Gelder, L. (1985, January). Sherry Turkle. Ms., pp. 48-49.

USING NEW TECHNOLOGIES IN EDUCATION: A REDEFINITION OF THE PROFESSION

By: Cecil Pretty, MICRO-IDEAS, Glenview, IL

The first step in our consideration of a redefinition of education as a profession is to establish a timeline which offers a sense of progression and change from past to future. Because of the rapid development of microcomputer technology, the period from 1975 to 1995 may be used to represent a window through which we can examine a very brief span in the history of American education.

PERSPECTIVES OF 1975: in that year many educators viewed the technologies with potential uses in education as follows:

* Faded Fads: in this group we can place phonographs, tachistoscopes, teaching machines and other devices related to the concepts of programmed learning;

* Undignified, Unprofessional, and Undisciplined: radio and television were two powerful media regarded with disdain by the "true" teacher;

* Lesser Evils: motion pictures, slide presentations filmstrips, and overhead projectors could be used by the "professional" - if considerable discretion was exercised;

* Inaccessible and/or Intimidating: teaching with videotape and computers was within the realm of possibility of very few teachers;

* What's That? interactive videodisc and satellite communications were virtually unknown;

* Tried and True: these were mainly the print media such as textbooks, workbooks, "dittos", picture books, coloring books, and "chalk and talk"; all those time-tested tools which nestled comfortably within the grasp of the traditional educator.

PERSPECTIVES OF 1983: here, almost at the center of our view, is the watershed of our terrain. Nineteen eighty-three has been called "the year of the report" when commissions, committees, and, it seems now, almost every group with vested interests (and in some cases, disinterests) took aim at everything we've done wrong. The aftermath leaves us with many problems and perspectives identified which, both directly and indirectly, point to the urgent need for change in education as a profession. Some of the issues are:

* the profession of education has been belittled and educators have suffered a loss of credibility;

* there is inadequate support for educator inservice and continuing education programs;

* increasing demands for accountability have resulted in an overwhelming emphasis on quantification. The time required to score students' work and to keep records now represents, in many settings, an unreasonable demand on teachers' time;

* techniques used to evaluate the competency and knowledge of teachers are often regarded as inadequate and/or unfair.

* financial restraints have resulted in excessive teacher-student ratios;

* the separation of the custodial and instructional functions of educational institutions represents a loss of opportunity and effectiveness;

* the conception and the execution of curriculum are often too widely separated. Teachers need and want more say about what they teach and the techniques used to teach it;

* the traditionally conservative nature of schools often precludes necessary and responsive adaptation to change;

* the dated styles and techniques of many school management and administration officials are increasingly seen as counter-productive and paternalistic. As a result teacher participation in decision-making processes is only a veneer.

PERSPECTIVES OF 1995: to prepare ourselves for this aspect of our examination, we need to consider a potential and reasonable configuration for a bundle of technologies which, at the present, are actually in relatively primitive states. Nevertheless, they are being used in education. And the microprocessor is at the heart of the bundle.

As the likely configuration of a computer-centered system of 1995 is presented, you should bear in mind that there is not much speculation involved in this proposal. In fact, the specifications outlined here may prove to be much too modest; most of the features outlined here are very nearly today's state-of-the-art. The important advances of the near future will be in increasing the levels of integration, compatibility, and interactivity, while achieving significant cost reductions.

What are the specifications of a typical commonplace 1995 system likely to be?

* inputs by mouse, joystick, touch screen and voice;

* the capability for the presentation and manipulation of outstanding graphics on screen and in print;

* powerful and flexible programs for word processing, idea processing, and worksheet processing;

* telecommunications made possible through satellite or cellular telephone technology with systems so powerful and efficient that it will cost more to monitor and invoice the services than it will to provide them;

* those communications systems, in reaching out to both span and link the world, will be capable of delivering information that will include the printed word, computer programs, music, and all types of still and motion video and graphics;

* individual computer memory will be so extensive that millions of bits of information will be able to be stored and manipulated by the hardware and firmware contained with a keyboard in a case no larger than today's lap computers;

* because of extensive networking capabilities, the distinctions between the micro and the larger systems will be blurred.

The impetus for realizing the potential for truly professional use of new technologies in traditional educational settings emanates from developments in the world of business and industry: the corporate setting.

Because of the rise of the industrial age in the Western world, since the late 1800's a strong challenge to the monopoly of publicly supported education has been gaining strength quietly but steadily. That force is corporate education. It has recently been described, in a Carnegie Foundation Report, as the "third leg" in the American educational system. These figures, as offered by TIME magazine (February 11, 1985), may help you gauge the measure of this activity:

	FOUR YEAR COLLEGES	CORPORATE EDUCATION
Approximate Numbers Of Students Served	8 million	9 million
Estimated Total Expenditure	$60 billion	$40 billion

Why has this happened?

(a) higher education has not escaped the barrage of criticism aimed at the lower levels of education. A major criticism is the watering down of academic curricula and standards in order to meet the perceived needs of the students being graduated from the school systems of America;

(b) new areas of growth and change in the American economy have resulted in a shift of career motivations for many college and university-bound students. Business related programs attract far more enrollments than studies in the liberal arts;

(c) institutions of higher learning, with their tradition-bound approaches to curricular change, faculty appointment and retention, and a perceived preferred detachment from real-world applications of research appear unable to adapt to rapid societal and technological change. A Texas Instruments executive focused on the dilemma by remarking recently: "as technology changes, universities tend to lag one to three years behind what's happening in the work place."

Then, we must examine how corporate education has, in contrast to public systems of education, been able to: 1) accomplish the adaptation to rapid change and 2) make use of educational technology for innovative and cost effective instruction. The following table presents some key elements relating to the offering of instruction in corporate and public education settings:

	CORPORATE EDUCATION	PUBLIC EDUCATION
Instructor's Contract	usually short-term and addressing specific objectives	mostly by tenure
Evaluation of Instructor	frequently and thoroughly	usually infrequent, often inappropriate
Employee Training & Education	increasingly given as part of salaried time (up to 15 - 20%)	generally inadequate and very often extra-curricular
Strategies for Instruction	prime concerns are for efficiency, access to resources, affective as well as cognitive outcomes	usually offered in lockstep, timeblocked inflexible groupings. degree of accountability varies widely
Uses of Technology	extensive--may include TV, telecommunications, tele-conferencing, satellite transmission, CBI, CMI, and interactive videotape or disk	much less than in corporate settings but increasing slowly

The striking differences are obvious here. But the comparison is quite unfair and should not be made without examining the philosophical foundations and the educational objectives of each of these systems of education.

Risking the pitfalls of speaking in broad generalizations, let me attempt to define succinctly the educative objectives of each.

CORPORATE EDUCATION is able to embrace, as a reasonably well-defined goal, the development of specific skills known to be more or less essential to bringing about an increase in the quantity and efficiency of commerce-related transactions.

THE K-12 CURRICULUM PUBLIC SYSTEM embraces the ideals of enabling the acquisition of skills, attitudes, and behaviors deemed as most germane, in a free and democratic society, to enhancing the quality of life.

THE UNIVERSITY AND COLLEGE SYSTEMS, as the hosts of the graduates of public and private school systems, are confronted with conflicting roles as to the emphasis required vis a vis research vs. practice. A fundamental question is being asked: should such institutions concern themselves with furthering knowledge or should they pander to societal and corporate pressures and offer "trendy courses".

When viewed in these terms of reference, the operation of many corporate educational enterprises with the rather simplistic, and often, single-minded objectives of such education, is by far, the easier task. But there are some important models to be studied, and lessons to be learned, from the corporate experience.

Bearing in mind that corporate experiences with its drive for efficiency and productivity, what are some definitive steps to be taken in developing a professional educator who is competent and confident in the use of appropriate computer-centered technologies? We need to examine the implications for change related to:

THE CURRICULUM: in order to adapt and utilize the computer and related technologies for educational applications a fundamental principle must be established: such technology should only be applied where it is most appropriate. If activities can lead to more effective learning without the use of technology then, so be it. But a great deal more research needs to take place in regard to the effects of computer use on achievement and attitudes as well as discovering the specific effects of using techniques such as branching and sequencing.

Because use of the computer-centered technologies greatly facilitate the individualization of instruction, the means of accomplishing curriculum objectives need to be expanded:

* teachers wish to take more control of their subject matter and its adaptation to local needs. Outcome objectives should be provided as guidelines which permit the customization of study as required by individual student needs, resources, and motivations;

* information-sharing, independent study, and discovery learning activities often do not fit well within the boundaries of regular timetables. More flexibility is required in the timetable: computers and telecommunications can extend the walls of the classroom as well as motivate students to become more creative, productive, and responsible for their own learning.

FOR TEACHERS AND TEACHING: Teachers are not prepared to use the new tools of the information processing society. They are comfortable with drill and practice which is often appropriate for the development of lower level cognitive skills. But teachers are generally not familiar with the testing of hypotheses, the manipulation of variables, complex educational games, construction sets, and similar responsive computer environments. The use of simulations, graphics generators, and statistical analysis procedures are not part of the basic repertoire of skills held by teachers. Students are often subject to performing tedious and mindless tasks instead of being stimulated to discover relationships, test theories, and explore and explain the unknown. The following actions must be taken in order to rectify this situation:

* teachers must be given the opportunity to learn the fundamentals of using the new technologies for the improved management and organization of their own classroom. Such instruction must be offered as part of their professional responsibilities. Teachers must begin to press for the inclusion of such activities being properly and regularly conducted as part of their working conditions. A significant number of teachers also need to become more professional in their attitudes toward participating in these activities;

* teachers must be provided with the opportunity to discover how computers are used in the real world, particularly within their own area of expertise. Unless teachers come to understand the power and excitement of using the techniques of information processing there is a reduced hope that they will come to understand the fundamental differences between the role of the teacher as a delivery system for instruction and the role of the educator as a facilitator and animator for student learning.

FOR SCHOOL ORGANIZATION AND MANAGEMENT: properly utilized as tools for improving the organization and management of schools, the new technologies can significantly reduce the present drain on human and fiscal resources while greatly increasing effectiveness. At the same time, a greater concern for the quality and quantity of human interaction must be developed. Indications are that, together with learning to use the new technologies for improving decision making skills, administrators must also be prepared to largely abandon the hierarchical model of management. A willingness to allow staff, students, and the public to participate in the decisions leading to a renewal of the educational enterprise has become absolutely essential. A current social theory claims that the use of high technology by society generates a greater need for "high touch" - a greater sense of caring and a deep concern for human values. School administrators need to assume that this is true and act accordingly; executive decision making must be collaborative. As for specific training for the professional administrator:

* if administrators expect to be effective as change agents in the instructional process and the teaching environment, they must also become knowledgeable in regard to the appropriate applications of educational technology.

* many administrators must also be taught the techniques of facilitating human communication. Education has long been concerned with the creation of structures which centralize, bureaucratize, and credentialize. Learning to function effectively and efficiently without such shields will, for some, be a very challenging assignment.

ABOUT OTHER RESOURCES FOR RENEWAL: even though the next few years will see major cost reductions in technological applications now considered to be beyond the fiscal resources of public systems, the required investment will be a significant burden to the public. It is believed that the resulting efficiency in the use of human resources and improvements in the standard of education will, in part, reassure the patrons of the schools that the sacrifice has been justified. If such an investment is not forthcoming, it is realistic to predict that current and future trends in politics and commerce could effect changes in American education that will further exacerbate the inequities that already exist in American society and education.

As one reviews the findings of the major reports on American education issued in 1983, several themes in common emerge. Two are particularly germane here:

* the schools and the community at large must combine to effect renewal in education. Bringing more effective education is seen as a broad scale social challenge;

* the decentralization of structures and local control of educational decision making by individual communities is seen as providing great impetus to effective action. Renewal will result from the innovative, creative, and imaginative solutions resulting from local efforts.

CONCLUSION: the impact of the computer on society has been monumental and most citizens are aware that virtually every human life in America is now affected by its use. Such a recognition will ease the way for the implementation, as never before, of new technologies in education. Unfortunately, there are many false hopes and expectations to be tempered and many unfortunate false starts have already been made. So, a word or two of caution seems appropriate here.

We must always bear in mind that the use of technology is not always appropriate; it can even be seen as harmful in some instructional settings. We must never forget that the use of technology is not likely to ever provide a satisfactory and humanizing range of tactile or auditory experiences for our children. And computers cannot really care for students nor be of much assistance in the development of interpersonal social skills increasingly demanded in a world of service occupations. And I will never remember a computer as fondly as I remember some of my human teachers. Will you?

CHILD TRACKING SYSTEM

By: Carolyn Quick, Illinois State Board of Education, Springfield, IL

This Child Tracking System was developed in response to requests from school districts for programs for a child tracking system utilizing microcomputers that would meet their needs for record keeping and have the capability of producing required Illinois State Board of Education child count reports.

As a result of these requests, the Illinois State Board of Education, Department of Specialized Educational Services has presently developed software programs for use on Apple II, II+, IIe and Radio Shack Model III and IV microcomputers with 2 disk drives and a minimum of 48K.

On a local level, the Child Tracking System will provide local districts and joint agreement special education administrators with a means for record keeping and retrieval which is useful as a tool for planning and decision making.

On the Apple II, II+ and IIe, there is a maximum capacity of 300 student records per diskette, however, it is recommended that you not exceed a maximum of 270 student records per diskette to allow for duplicate records that may be necessary to distinguish separate fund codes throughout a school year. On the Radio Shack Models III and IV, there is a maximum capacity of 350 student records per diskette with a maximum of 280 student records per diskette recommended.

Each student record will contain two screens of information about the student. The Apple and Radio Shack formats vary slightly but contain the same information. The individual screens (Apple Models) are simulated here:

SCREEN #1

STUDENT INFORMATION

```
NAME:                                                          TERM:
          (LAST)              (FIRST)          (M)
ADDRESS:
PHONE:  (   )  -                                  BIRTHDATE:  /  /
PARENT/GUARDIAN:                                                SEX:
CHAR/SEVERITY CODE:                                        ETHNIC CODE:
LANG CODE:     PROGRAMS/SERV TYPES:
DISTRICT ID:                              STATE ID:
RESIDENT DISTRICT:
CLAIMING DISTRICT:
LEAST RESTRICTIVE ENVIR:                                   POST HOME:
STAFF CODE:                                                FUND CODE:
REFERRAL TYPES:
RELATED SERVICES:
PERS/STU RATIO:                                 OTHER AGY:
STATE OP FACILITY:                                 PRIV. FACILITY:
EDUCATION COST:  $                                   TRAN COST: $
                    USE RIGHT ARROW FOR NEXT FIELD
                     USE RETURN TO FINISH RECORD.
```

SCREEN #2

STUDENT INFORMATION

/ / BEGINNING DATE
/ / DISCONTINUED DATE

/ / REFERRED
/ / EVALUATED
/ / PARENT NOTIFIED
/ / STAFFING
/ / MULTISTAFF CONF
/ / I.E.P. COMPLETED
/ / DISTRICT APPROVAL
/ / I.S.B.E. APPROVAL
/ / I.S.B.E. REPORT
/ / ORIGINAL RECORD
/ / LAST MODIFIED

COMMENTS

USE RIGHT ARROW FOR NEXT FIELD
USE RETURN TO FINISH RECORD.

The system has validity checks built in which will not allow continuation of input until valid information has been entered. However, as in any computer system, it cannot guarantee the accuracy of information it cannot validate.

After the original student records have been entered, the records may be added to, updated or deleted at any time.

Once the student and teacher information has been entered, it is possible to print reports from this information. It is possible to print the I.E.P. first page, class lists, annual review lists, three year evaluation lists, frequency reports and cross-tabulation reports.

The Child Tracking System can be used to provide the Illinois State Board of Education and the Department of Specialized Educational Services with electronic reporting of information via diskettes for both the Chapter I, 89-313 Child Count submission and the P.L. 94-142 child count submissions for FACTS. In this system is a disk check program which will check for missing or illogical field entries, generate a printout of the information being sent to the Illinois State Board of Education and produce diskettes to carry these reports to the Illinois State Board office for processing.

A final option of the Child Tracking System provides an access code for writing and managing local programs that can interact with the Child Tracking System program.

If you are interested in obtaining the programmed diskettes and appropriate manual, complete the enclosed form and send it with your diskettes to the Illinois State Board of Education.

CHILD TRACKING SYSTEM

Date: (DATE)

I would like a copy of the Child Tracing System for:

(check one)
[ ] Apple II, II+, IIe, 2 disk drives, minimum of 48K
[ ] Radio Shack Models III and IV, 2 disk drives, minimum of 48K

Enclosed are:

(check one)
[ ] 4 diskettes for the Apple version
[ ] 2 diskettes for the Radio Shack version

Please send the programmed diskettes and the appropriate manual to:

Name: (NAME)

Organization: (ORGANIZATION)

Street Address: (STREET)

City: (CITY)

State: (STATE)

Zip Code: (ZIP)

Return to: Marcia Sailsbury
Special Education Specialist
Illinois State Board of Education
100 North First Street
Springfield, IL 62777

COMPUTER LITERACY: IS IT STILL AN ISSUE OR SHOULD WE CHANGE ITS NAME?

By: David Rine, Western Illinois University, Macomb, IL

Introduction

One of the most widely discussed topics during the last several years in education has been "computer literacy". In a 1982 study, the National Center for Education Statistics reported that schools used computers to teach "computer literacy" more than for any other computer related activity, yet many definitions abound. In the 1970's the term seemed to imply some knowledge of programming. But since then the definition, as a fast moving target, has changed dramatically. Moreover, in a MICRO-IDEAS conference recently in Chicago, Illinois, Sylvia Charp, Past President of AFIPS, and David Rine, a Past Chairman of the IEEE Computer Society Education Committee, suggested that it may be time for the term to die out and change to a more descriptive term, reflecting the need for students to learn to deal with many emerging technologies that are not all related to the computer, for example telecommunications.

In this short write-up, we ask the following questions and offer some answers in outline form:

- Q1(a) -- Has the expectation that computer literacy include computer programming changed in the last six years?
- Q1(b) -- What changes in "programming ability" are seen in the next few years?
- Q2 -- What are the forces behind the changes in computer literacy?
- Q3 -- What should be the balance between concepts and skills in a computer literacy course?

COMPUTER LITERACY

Definition

Computer literacy may include emphasis on learning about computers as opposed to learning through computers or learning with computers. These viewpoints are characterized as follows:

Learning about computers --
- Prelude to computer science, data processing, ...

Learning through computers --
- Drill and practice
- Diagnostic testing
- Tutorial programs

Learning with computers --
- Simulations and games
- Problem solving
- Creative activities

A dictionary definition of computer "literate" would include:

1. can perform basic skills;
2. knowledgeable about computers;
3. familiar with or well-informed about computers; and
4. can read and write about the subject.

Questions

My questions then are:

1. What skills are necessary (to work and live in an Information Society...)?
2. What concepts are necessary (to understand, adapt and make decisions...)?
3. What terms are necessary (to write and read...)?

Some complicating factors are:

1. certain skills (such as the use of tools) change with time (e.g. 1970's through 1980's);
2. standards (e.g. terms) tend to change with time (though we need standards and should avoid changing too fast); and
3. understanding of concepts also tends to change (better or worse).

COMPUTER LITERACY AND PROGRAMMING

Answer to Question One(A)

The concept of "programming" has changed dramatically, from that of language and short programs to that of development tools and larger programs. Still, in its essence, a program -- and hence programming -- may be characterized as follows:

Program -- solution to a problem
Problem -- where you are (current state, eg.g. on the roof)
Solution - where you want to be (new state, e.g. on the ground)
Program (algorithm) -- how to get from current state to new state (hopefully safely, reliably, efficiently, understandably...).

There is a new emphasis on training the end-user of computers rather than just the computer programmer. The concept of "using information and service resources" has changed dramatically. There is now, for example, more emphasis on ethics, laws, security, integrity, impact of errors, timeliness in the way we do business, rapidly prototyping models, and simulations of new possible ways of doing things in sciences and professions.

Answer to Question One(B)

Anticipated changes in "programming ability" requirements include:

1. knowledge of preferred tools (editors, debuggers, prototypers, verifiers, ...) for certain applications;
2. ability to function in and control a computer environment, with emphasis on systems design, abstraction and classes of applications (this implies less emphasis on simply knowing language syntax); and
3. ability to use software engineering techniques.

Also, computer educators will need greater understanding of what tools are preferable for a given level of programming learner, where level is affected by age, mode of learning, current knowledge, and so on.

FORCES BEHIND THE CHANGES

Answer to Question Two

I would agree with Andy Molnar of NSF, who compared the difficulties of defining "computer literacy" with "aiming at a moving target," and would say that one difficulty is that we must be careful in our educational system not to create a 2-class culture: the computer literate and the computer illiterate.

The reasons for changes in the meaning of "computer literacy" include:

1. human creativity -- the more we know the more we want to know;
2. survival -- of business, governments, cultures;
3. productivity and convenience -- search for better and shorter ways;
4. basic needs to understand our universe and origins; and
5. power and other psychological reasons.

CONCEPTS VS. SKILLS

Answer to Question Three

In a field in which the specific equipment on which students acquire skills may become outdated by the time they graduate, the following considerations are relevant:

1. cost of new equipment;
2. cost of personnel time needed to implement and document new equipment;
3. leadership in requiring experimental use of new equipment; and
4. accountability for teaching students how to start using new equipment.

A balance is needed between simply teaching skills (e.g. typing on a word processor) and teaching the use of those skills to improve the student's situation in life (e.g. jobs for unemployed and minorities, ...).

Conclusions

Topics

The following topics should be included in a Computer Literacy course:

1. computers as a (personal) tool;
2. computer components and functions;
3. communicating problems and solutions with computers;
4. (personal) computer applications;
5. advantages and disadvantages of (personal) computers; and
6. impact of (personal) computers on the individual (and society).

Approach

The approach to teaching a computer literacy course should emphasize:

1. concepts;
2. skills;
3. classroom;
4. lab/hands-on; and
5. resources.

NEW STUDIES

1. National assessment of educational progress: Assessment of computer competences. E.T.S., Princeton, NJ 08541, 800/223-0267.

   Assessment -- 1985
   Results -- 1986
   (Schools)

2. EDUCOM computer literacy project. EDUCOM, P.O. Box 364, Princeton, NJ 08540.

   Start -- 1983
   Implementation -- 1986
   (Schools and college)

IMPROVING INSTRUCTION THROUGH THE USE OF APPROPRIATE SOFTWARE FOR SPEC. ED.

By: Penny Shattuck, MICRO-IDEAS, Glenview, IL

As microcomputers become more prevalent in the classroom, the job of finding the software that works is becoming more difficult. Teachers do not have time to preview all that is available and without a lot of time and experience it is often difficult to determine exactly what a software program is capable of doing. Add to that trying to find the program that fits into the curriculum and fits the individual needs of each child and you really have a challenge. The following programs are utility programs by MECC. These programs allow the teacher to enter their own data. The programs are: a list of criteria for Special Education software. One program that fits most criteria for good software is Logo in the classroom.

CRITERIA FOR LD SOFTWARE

Children with learning difficulties have particular learning needs, therefore software used with these children should address their needs. Below is a list of specifics that should be incorporated in effective special education software.

1. All responses should be consistent, i.e.: Y for yes throughout the program.
2. The number of variables and concepts presented is appropriate for the ability level of the user.
3. Content progresses from easier to more complex.
4. Criteria for completion of the program are established and made known to the user.
5. Enough positive reinforcement is provided to maintain a high level of motivation.
6. Teachers are provided feedback on how well their students have done.
7. Students using the program will receive a sufficient percentage of responses correct.
8. It is possible for a user to go back and re-enter a program at any point.
9. Prompts and hints are provided when needed.
10. The user can leave a program at any point, then re-enter at the appropriate point.
11. Inappropriate auditory feedback is avoided.
12. Feedback to a student response is always provided.

13. Automatic re-teaching after a specified number of wrong responses provided.

14. "Cute" screen display which delay or distract from the purpose of the program are avoided.

15. When instructions appear on the screen they stay on until the user is ready to move on.

16. Instruction may be bypassed when not needed.

17. Presentation of extraneous information (copyright, author, etc.) may be bypassed.

18. Signaling system (arrow, flashing cursor) stays on screen as long as needed.

19. If the user makes a response error, he/she is able to correct it.

20. Feedback to student responses stays on the screen until the user is ready to move on.

21. Visual displays are clear, easy to read (larger print), and uncluttered.

22. Student responses are simple and easy to enter.

23. User signal to continue is consistent within the program (always space bar, return, etc.).

24. The program is "crash" proof. Unexpected input will not cause problems.

25. Sound can be deleted at the option of the teacher.

26. Appropriate materials are provided with the program which include necessary information for the teacher, i.e.: objectives of the program, instructions for use, appropriate grade level, etc.

27. The objectives of the program match the instructional objectives of the district/school/class.

28. The objectives have educational significance.

29. The programs either does or can be adapted to fit the curriculum objectives.

30. Branching should be incorporated whenever possible. Keeping the level of difficulty to the student's level so that he/she may participate without frustration.

31. Students can complete program in available time.

32. Skills are broken down to subskills allowing the student to build on what they learn.

33. Appropriate use is made of graphics/sound/color.

34. The program is motivational.

35. Students are provided feedback on how well they are doing.

36. At the end of the program students are provided feedback on how well they have done.

37. Teachers can easily implement the program.

38. Learners can operate the program independently.

39. The program is reliable and accurate.

40. The program makes effective use of computer capabilities.

WORD WIZARD

WORD WIZARD is divided into four programs. The first is Cast A Spell. It is a drill and practice program which provides spelling exercise. It should be used with small groups ranging from 1-6 grades. OBJECTIVES: improve spelling on a list of words; improve word recognition.

The program starts out by telling the student to put his data diskette into the drive. He starts the program with selecting one of the lists by a number. Directions are given and the student may decide how many words he cares to work with. There is a maximum of 15 words. They may be no longer than 9 characters. At the end the score is given and the missed words are listed. There are handouts to go with the program and the teachers may create their own list.

The second program is Witch Word. It is a word game that presents a scramble of letters to be reordered to form a word or words. Each letter must be used. The grade range is 2-6th grade and it should be used with an individual or small groups. OBJECTIVES: improve in ability to order letters to form words; recognize the combinations of letters which form words or parts of words. Graphics are a large part of this program. The directions are given by two witches. The students then try to unscramble the letters and make the word the program is looking for. At the end the score is given. There are handouts to support the program.

The third is Wizard Magic. It is a word guessing game in which students fill in dashes with letters to make a particular word. Hints to the words are the number of dashes and the presentation of a clue. It is an educational game that deals with language and logic and should be used with grades 2-6 in individual or small groups. OBJECTIVES: associate words as composites of letters; infer words from definitions; make logical guesses.

The student may use an existing list or make up a list to use with a friend. There are 12 words on a list and a student may pick any number to work with. The program may be played with one or two players and their names are used. There may also be clues if desired. When students input their own word, no

clue is given. There are worksheets available.

The fourth program is Wizard Memory. The students begin by choosing a list of words. Students are given all too brief instructions. It is suggested they use antonyms and examples are given. The program can be played by one or two players. On each turn, a player selects two stars by typing in the letter of the star. The object is to choose stars which contain the same words, sets of homonyms, antonyms, or synonyms, or other word pairs. When the game is over the scores are given and if two are playing the winner is identified.

STUDY GUIDE

STUDY GUIDE consists of three programs: DESIGNER, REVIEWER, and EXAMINER. The DESIGNER program enables you to create, edit, and store sets (lists) comprised of multiple-choice, matching, completion, and true-false questions. You may specify objective statements and numbers for the questions that you write. You may also write remedial feedback that students will see if they choose the incorrect answer.

After you create the sets with the DESIGNER program, students use the REVIEWER program for interactive study at the computer. If you wish, you may direct students to review questions by specific objective categories. The EXAMINER program lets you print the sets that you created as tests or worksheets. You may specify the number of questions to print from each objective category.

Before you use the DESIGNER program for the first time, you will need to format a data diskette using a blank or recycled diskette. The Diskette Support option on the Management Option menu will help you prepare your data diskette and set up the disk drives. You may access the Management Option by typing Control A when the main menu appears. To do this, hold down the Control key and press "A". The Management Options menu will then appear. The DESIGNER program, as well as the EXAMINER program, will be on this menu.

This program enables a teacher to:

* create and edit sets of multiple-choice, matching, completion, and false questions and answers;
* group questions by objective statements and numbers;
* write remedial feedback that students will see for incorrect responses;
* store sets on a data diskette which can be retrieved for interactive student review with the REVIEWER program;
* store sets on a data diskette which can be printed as worksheets or tests with the EXAMINER program;
* review students' scores.

SCHOOL UTILITIES VOL. 2 (READABILITY)

Almost all readability estimates use some measure of word difficulty and syntactic complexity as their main predictors. Examples of word difficulty predictors are word length, syllable counts, and the number of unfamiliar words. Examples of sentence difficulty predictors are the number of dependent clauses and average sentence length. Of these, the number of unfamiliar words and average sentence length have proven to be the best predictors.

The programs are arranged in the following order:

INSTRUCTION
ENTER TEXT
EDIT TEXT
READABILITY TESTS (6 tests: Spache, Dale Chall, Fry, Raygor, Flesch, Gunning-Fog)
DISKETTE MAINTENANCE (catalog, list passages, remove text from diskette)

****

The best test to use for fourth grade and below is the Spache method.
The best test to use for fifth grade on up is the Dale Chall method.
The Fry and Raygor methods should be run along with both of the above as a cross check.
****

TO SELECT SAMPLE PASSAGES

1. For most books 3-5 samples should be enough.
2. Use sample passages of about 100 words each. The program prints out running total at top of screen.
3. Select a passage near the front of the book, but avoid first paragraph. Other passages should be from the middle and the end. But not the last paragraph in the book.
4. Choose sample passages that are representative of book as a whole.

ENTERING TEXT

1. Precede each instance of a proper name or any word you want the computer to treat as a familiar word with a pound sign.

2. DO NOT include periods with abbreviations or as the decimal point. This will give a reading of less difficult than it really is.

3. Spelling errors must be corrected. This would make the reading more difficult than should be.

4. Hyphenated words are counted as one word.

5. Ellipses (...) are counted as one (end of sentence) mark.

6. The following punctuation marks may be entered normally, but will be ignored in the analysis: ():',

****

The READABILITY TESTS program permits the selection of one or all of the tests to be applied to passages of text. It is best to run all tests when:

1. the readability level of text is doubtful.

2. comparison of test results is useful.

3. the program user is unfamiliar with the individual tests.

You do have a choice of one or all tests. Other choices given are having both the text and the individual words of the text with syllable counts displayed on screen and/or from a printer.

After you have used the readability test on passages it is possible to delete the passages. It is also possible to run the program on a one or two disk machine. These selections are all menu driven.

****

USING LOGO WITH LD STUDENTS

LOGO is a user-friendly computer graphics language based on turtle geometry. It consists of a relatively small number of commands and rules but it is also open ended. One of the reasons for the success of Logo is the student is trained to find out solutions for himself. By entering a sequence of commands, a student can communicate with the imaginary turtle on the screen. There is no one way to answer a problem but many ways. No less important is the socialization which goes on when two children are working on the computer together. It is also an excellent way to develop an intuitive understanding about what computer programming is about.

For all these reasons and more, a Logo lab is an excellent environment for mainstreaming. There are, however, some variables which add to the success. The amount of positive social interaction and peer support that can be expected is likely to be greater when LD and non-LD students are learning novel and motivating material such as LOGO rather than reading or mathematics. Some of these are:

* The instructor should be aware of which students are LD students.

* The ratio of computer/user should be no more than two students.

* A variety of modes of presentation should be used to introduce basic concepts.

* Physical activities involving the concepts of directionality, distance, should be used occasionally.

* Let children role play the turtle using squares to help count number of steps.

* Poster, chalkboards and flashcards should be available to help students remember the commands.

* Lectures, demonstration, discussion, and hands-on activities should be evenly distributed throughout the class period.

* Extra reading assignments should be prepared and given to LD students for follow-up practice at home.

* Any questions should be multiple choice rather than open ended.

* Demonstrating the procedures first and then explaining the concepts and processes involved.

* Separate the children from the computers while giving a verbal lesson.

* Be insistent that students write down their ideas for solving Logo related problems and check them over before keying them into the computer. This reduces play time and gives the student a copy of the commands and statements used.

TAKING THE BYTE OUT OF COUNSELING

By: Jane A. Simmons, Maine Township High School East, Park Ridge, IL

INTRODUCTION

First, all of the forms and material and information maintained are on an IBM compatible and IBM PC. This is not connected to the school district's main frame.

There is no doubt that this makes the use and application of a computer easier: there is no log-on for the mainframe, and no need to go from a software program to the district's files.

The vast majority of the work was done with PFS FILE, REPORT, and WRITE. These were chosen because of their extreme versatility and ease of use. A few of the letters were done on WORDSTAR employing MAILMERGE. However, all could be done using the PFS series.

MY COMPUTER AND I:

Every counselor is in large part a record keeper, and expected to track a student's performance, and behavior. Along with this, one must communicate with the faculty, the parents, and the students themselves, write recommendations for college and for scholarships.

In this era of accountability, schools document everything, and the counselor is the end recipient of these pieces of paper; we are expected to know the progress (or lack of) of each of our counselees, and then often relay this in another form to someone else. To make all of this easier, my computer and I have become inseparable.

The most beneficial end of this presentation is to provide you with samples of what can be done. Because of the versatility of the PFS FILE series, which allows you to form any type of file and information sheet you wish, the forms and files can be shaped to individual needs. What follows is a list of the various forms, files, letters and uses I have found helpful.

A LITTLE BACKGROUND ON THE PFS SERIES AND WHAT YOU CAN DO:

My choice in a software system was (1) ease of use (2) wide versatility in the format the files could take and (3) integration. The PFS SERIES meets these requirements.

FILE: This program allows the user to create, beginning with a blank screen, any type of "form" he wishes to employ, just as you would begin with a blank sheet of paper. Any category created can later be the basis of retrieving information or collating information. Any category can also be used in a form letter created with WRITE.

REPORT: This program allows the user to generate lists from the FILE. Any category in the FILE can be the basis for retrieval, using also a "more than" and "less than" and "missing" category. The report hardcopy can be organized with any category from the file listed first. (Examples follow in the sample form section).

WRITE: This is the word processor part of the integrated series. This coordinates with the FILE program and allows the user to create form (mail merge) letters pulling information from any file and any file category to create individual letters. Needless to say, this is also the "electronic typewriter" of the series. It is not as powerful as WORDSTAR, nor will it let you do mailmerge letters as easily, but it is very simple to use.

STUDENT INFORMATION AND FILES:

I. INFORMATION FILE: This records the names, addresses, phones, parents' names of all my counselees. This information helps me when I need to contact the parents, know whether a student lives with one or both parents, and if I want to write a form letter, I have the names and addresses already finished.

RECORD: This records incoming test scores, college tests, and credits earned under the categories of courses and those needed for graduation. For state mandated courses, I can record whether a student has passed the course. Using REPORT, at any given time, I can generate a list of those who need various courses to graduate, or who hasn't taken Drivers' Education or whatever.

NOTES: I receive a copy of every failure notice, behavior report, and other extraneous notes sent on the students. I record every piece under the student's name. At any time I can print this out and check the progress, confirm if a failure notice was mailed, and so on. I am required to communicate student information to the teachers of those students in special programs. This form is readily printed and conveyed to Special Education coordinators and teachers of English as a Second Language.

These three compose the core of the information I record on each of my counselees. Granted, the first year of use, one must enter this for all four years, but after that, one is only entering the freshmen. (The NOTES are done during the year on a daily or weekly basis, depending what there is to write down.)

II. INFORMATION RELATED TO COLLEGE

One area of our concern is writing letters of recommendation for seniors applying to various schools, writing scholarship letters, and informing the kids of scholarship opportunities.

LONG-FORM APPLICATION INFORMATION SHEET: I begin by having my seniors complete this so I have the background information with which to write a letter of recommendation.

LETTER OF RECOMMENDATIONS: I use WRITE for these, along with a PFS file for each school each student will attend (called COLREC for College Recommendation information). This allows me to do an individual letter for each school. (I still feel, as my mother taught me, it's the looks that count. I can't believe that a Xerox letter conveys interest and concern on my part.) It is here that I employ another PFS program, called PFS PROOF. This program is a spelling check program. Later, I can alter the letter a little bit, and employ it for any scholarship.

SCHOLARSHIP FILE: I ask the students to write down their ethnic background, religion, etc., and I record it on the file. When a scholarship comes through for "those of Greek-American Indian descent," or "those whose father works in

a plumbing factory," I can use a WRITE letter, and retrieve the names from the FILE based on just this information and inform the students individually of scholarships which may be applicable to them.

PARENTS' LETTER: This year I decided that I had to make a concerted effort to talk to the parents of my academically superior and capable students to make sure that the entire family understood the application process, financial aid, deadlines, and so on. So, I wrote a personal form letter inviting the parents to come in and talk. However, I wanted to make one paragraph a bit different depending upon the rank of the student. So, first I used REPORT and the RECORD file, and pulled up the students according to class rank (1 to 50, 51 to 100, 101-200, 201 to 370 [half the class]). There is no doubt that the rank and the possible school acceptances are correlated. I then wrote one letter, copied it on the disk, and just changed the one paragraph relating to the achievement, rank, and possibilities. I then merged this so each parent had an individual letter. (Just to satisfy your curiosity, I mailed about 51 letters, and all but 3 parents have been in for a conference.)

III. MEMOS AND LETTERS

LETTERS TO STUDENTS: I enjoy commending my students and letting them know that I am aware of their achievements. Before a computer, it was time consuming and difficult to write each one. Our faculty now sends out letters of commendation, and each faculty member can also name a "Student of the Month"--the one outstanding student. I have a form letter corresponding to each of these, and all I have to do is enter the name of the student, and the class.

I also have form letters congratulating individually my students who are elected to National Honor Society at various times.

MEMOS: Through our homerooms, it is convenient to make announcements concerning registration, testing, etc. I now have my disk file of "annual" announcements, and I need only change a few items, and then reprint, instead of retyping the entire memo. What I have found, most importantly, is that I communicate much more efficiently and better because every memo and every note is not a chore.

LETTERS TO THE FACULTY: I often have to write to all the teachers of a student, and the mail merge lets me do an individual memo.

We have an absence policy at our school, and when a student brings a doctor's note, the days do not count toward a potential drop from the class. I have on a file a doctor's note, an ill note, a religious holiday note, and a death note. All I need to do is fill in the names, and each teacher gets an individual letter.

IV. ACCOUNTABILITY: This fall our district suggested that counselors record their daily activities. Instead of a check-off list on paper, I created a FILE form. Daily I entered the names of the students, parents, teachers, and administrators I had seen or talked to, and the amount of time I spent on each task. I then used REPORT to tally the daily results.

V. CLUBS: As a club sponsor, I need to communicate with my club members. I have a file of their names, their homeroom teachers' names, their addresses, etc.

Finally, perhaps the biggest advantage of using a computer is the ease of altering information and ease of changing letters for wider application. Although there is much individual work in counseling, many letters and forms are repeated, but need new dates, etc. A computer lets you retrieve a letter from the disk, alter the information, and then print it out. I have found that I communicate more frequently and more because it is so easy.

****

NAME: GRADE: STUDENT #:

DOB: SS#: SEX:

HR CODE: HR TCHR: ROOM:

PARENTS:

ADDRESS:

CITY: STATE: ZIP:

H PH: MO WK PH: FA WK PH:

COMMENTS:

NAME:

8TH GRADE TESTS--LOCAL % AND STANINE:

READING: MATH COMP:
WRIT SKILLS: MATH CONCEPTS:
SOC STUDIES: SCAT VERBAL:
BIOL SCI: SCAT QUAN:
PHYS SCI: SCAT TOTAL:

PSAT--OCT. 85: VERBAL: MATH: SEL SCORE:

ACT: DATE: ENG: MATH: SS: NS: COMP:
DATE:
DATE:
DATE:
SAT: DATE: VERBAL: MATH: TOTAL:
DATE:
DATE:
ACH: DATE: NAME-SCORE: NAME-SCORE:
NAME-SCORE: NAME-SCORE:

AP TESTS--DATE: NAME-SCORE: NAME-SCORE:
NAME-SCORE: NAME-SCORE:

RANK AND GPA: CLASS SIZE:
4TH SEM: GPA:
5TH SEM: GPA:
6TH SEM: GPA:
7TH SEM: GPA:

CREDITS EARNED:
ENG: PE:
MATH: HEALTH:
SS: CONS ED:
SCI: DR ED:
FOR LANG:
BUSINESS:
FINE ARTS:
HOME ED: TOTAL ACADEMIC CREDITS EARNED:
IND ED:
WORK PRG:
SPECIAL ED:

SEQUENTIAL RECORD OF REPORTS------1984-5 SCHOOL YEAR

NAME: HR CODE:

BR=behavior reports; BCH=behavior calls home; F=failure notes; NF=near failure notes; NF/F?=category not marked; RPT=report to parents; DN=doctor's note; RN=religious note; A=absence letters; GN=commendation note; DA--DS=detentions assigned-detentions served;

FALL, 1984

TO: ALL SENIORS

FROM: Dr. Simmons

RE: Scholarship Information Form

Many scholarships are sponsored or offered by professional groups, schools, organizations, and businesses for those planning on a college career. For example, some are offered for vocations in a health, teaching or secretarial career.

Other scholarships are offered for students living in certain areas, or interested in some activity, such as a sorority or fraternity. Others are available to students based upon ethnic or religious background. Still others are offered through the colleges themselves.

This year, to insure that you receive notification of all the scholarships made available through Maine East and for which you are qualified, I am supplying an information sheet to you.

If you supply the information as outlined on the sheet you will be notified personally of any scholarships applicable to you based upon the information you supply. It is important to notify me of any changes in the college choices or major.

FULL NAME: FIRST NAME:

SEX: HR CODE: HR TEACHER:

COMPLETE ADDRESS:

RANK SEM 6: SEM 7: US CITIZEN: PERM RES:

INTENDED MAJOR: RELIGION:

VARSITY SPORTS: TALENT:

ETHNIC BKGRD:

UNION MEMBERSHIP OF FATHER:

UNION MEMBERSHIP OF MOTHER:

COLLEGES:

HR TEACHER

PLEASE DELIVER

TO: *FULL NAME*

FROM: Dr. Simmons

RE: AVAILABLE SCHOLARSHIP

Based on the information you supplied to me, it appears you may want to apply for the following scholarships:

These are available in the Guidance Center. Do not hesitate to obtain the application and begin work on it. The deadline may be quite soon.

Remember to pick up the scholarship(s) now from the career counselor in the guidance center.

****

MRS. LAOURAS

PLEASE DELIVER

TO: Jane Chen

FROM: Dr. Simmons

RE: Available Scholarship

Based on the information you supplied to me, it appears you may want to apply for the following scholarships:

JOSTEN'S SCHOLARSHIP

These are available in the Guidance Center. Do not hesitate to obtain the application and begin work on it. The deadline may be quite soon.

Remember to pick up the scholarship(s) now from the career counselor in the guidance center.

SWEETENING YOUR APPLE

By: Ralph Sullons, John F. Kennedy Jr. High, Rockford, IL

The computer classroom should be a place for the student to think about, explore, and solve problems. One of the best things about a computer is that it helps you think about, explore and solve problems in an interactive way. The Apple computer allows you to use various input and output devices, which make for interesting ways to solve problems.

Topo was one of the first robots made for a microcomputer (Apple Computer). It sold for about $500 and was controlled by an Apple computer. In the beginning all it could do was move and turn. It worked using Logo, BASIC, TopoFOURTH, and machine languages. Later models had many added devices--voice, sensors, appendages, etc.

Topo, a robot, as a tool can help students learn to solve problems in the computer classroom. In the future the ability of our students to solve a problem by manipulating and interpreting data will make the difference between successfully competing or failing to compete in life. Topo helps in the teaching of the understanding of computer languages and problem solving by giving the student a concrete representation of an abstract idea.

In my class I use Topo to introduce flowcharts, Logo and the introduction of program languages. In BASIC Topo uses subroutines which make it easy to use in BASIC and lets you make up your own language (see example 1). In the making of flowcharts in class, I use Topo and Cricket to test student's problems to see if their flowcharts work. Students write flowcharts for a robot that make a robot (them at first), do different things as--walk around a chair, walk to a wall and back, or walk to a corner of the room. The Cricket tells them what to do when they are the robot and they tell Topo what to do when they are in charge.

The Cricket Voice Synthesizer is one of the most useful peripherals you can buy for your Apple computer. It gives you three kinds of speech (a natural female voice output, an unlimited vocabulary robotic voice, and phoneme code), sound effects, stereo music, plus a clock. If you have an Apple IIc all you do is plug it in. The Apple IIe requires an extended memory card and either the Street Electronics Alphabits card or an Apple Super Serial card. It uses PRODOS and comes with a disk of sample programs.

The Apple Mouse facilitates the use of the computer in the classroom by allowing both teacher and student to process and use information in a quickly and easily learned way without being programmers. Programs, such as the new Bank Street Writer, Story Maker, How About a Nice Game of Chess?, Dazzle Draw, Grandma's House, and Mouse Paint, are only the start of an all-new generation of programs which use Mouse and/or pad technology.

The future of computers will not be just in the world of programming, but a world of robots, voice synthesizers, keypads, Mouses (or is it Mice?), and sound. At first the use of computers in schools was mostly for teaching programming; the next move will be toward using programs and peripherals as a tool to problem solving.

EXAMPLE 1

```
10 GOSUB 7000 :REM INIT
20 HOME
30 INPUT " ";C$
40 IF C$="WALK" THEN N=20 :GOSUB 5100 :REM TFD 20
50 IF C$="TURN" THEN N=90 :GOSUB 5200 :REM TRT 90
60 IF C$="TURN" OR C$="WALK" THEN GOTO 30
70 PRINT "NOT A ROBOT KEY WORD":GOTO 30
```

(SEE ROBOT FLOWCHART WORKSHEET)

SAMPLE OF MICROCOMPUTER USES IN THE EDUCATION SYSTEM

Drill and Practice
Tutorial
Materials Generation
Simulation
Computer Programming
Management Systems (class and office)
Test Scoring and Analysis
Computer Awareness
Data Base
Office Use
An Aid To Instruction
Games

WHY USE A MICRO?

I. To MOTIVATE
II. To provide for individual differences
III. To build desirable attitudes toward learning
IV. To reinforce learning with instant feedback
V. To provide skill maintenance
VI. To provide individual and appropriate test and homework for students
VII. To provide a variety of techniques in the classroom
VIII. To provide an enjoyable way to practice concept development and skill building

CHARACTERISTICS OF GOOD EDUCATIONAL SOFTWARE

I. User Orientation
   - A. Menu Driven
   - B. Flexibility (uses, settings, levels)
   - C. Freedom from Need of Assistance
   - D. Freedom of System Errors
   - E. Compatibility with other materials used

II. Classroom Tested
III. Supported by a good Teacher's Manual:
   - A. In an Understandable Language
   - B. With Sample Runs
   - C. Commentary
   - D. Students' Records

IV. Features good use of Graphics and Sound to motivate
V. Features a Concept Development, nurturing of Logical Thinking and Problem Solving Strategies
VI. The Computer should play an Unique Role

THE COMPUTER AS A TOOL IN THE JUNIOR HIGH CLASSROOM

By: Frederic Tarnow, Keller Junior High School, Schaumburg, IL

For too many years now, because of budget restraints, science teachers have had to put up with the lack of equipment capable of making accurate measurements of time, distance, temperature and light. The results of not having such equipment has been the acceptance of error factors in the range of 25 percent of more as we have had to rely on estimation, simple thermometers, hand held stopwatches, meter sticks observations and judgment calls. These error rates were further complicated by students mathematical manipulation of the collected data (often at a level which was beyond their abilities). After which we often must spend hours explaining why reality and their results are different.

Fortunately with the availability of the microcomputer in ever increasing numbers in our schools, we now have the capability of utilizing a high tech piece of equipment which can accurately measure such factors in our labs within error rates of between 1 and 5 percent. The best part is the fact that the cost of additional hardware and software necessary for turning the microcomputer into a laboratory tool is within most of our school's science budgets.

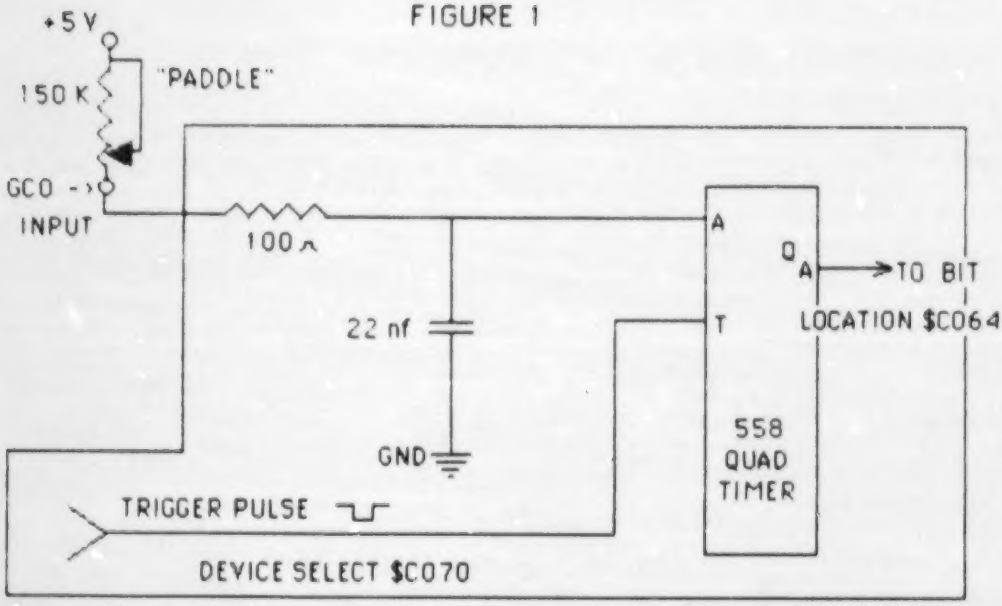
Because of the fact that the microcomputer has the ability to react to changes in resistance, the game paddle I/O (i.e. internal connection on II, II+, and IIe) provide a simple but powerful way to directly input laboratory measurements into the computer. One might well think by the name "game paddle," and the relatively low performance capability of its electronics that there would be little, if any, practical applications in its utilization. However, there currently exists a wide range of science experiments in all areas that can be applied from fourth grade through college which make use of game paddle inputs. To date so far, the vast majority of both hardware and software has been developed utilizing the Apple game paddle circuitry. This is not to say that other brands of microcomputers could not be used to accomplish the same task.

The most important part of the Apple's game paddle electronics, with regard to utilizing the computer as a tool in a science lab, is its set of four analog-to-digital converters. These are measuring devices that convert an external resistance into a timed signal that the computer can measure with the aid of appropriate software. Since the game paddle is in effect, an external variable resistor with one end connected to a 5v. power source, current flows through that resistor charging it with the capacitor inside of the Apple.

A circuit diagram of the game paddle input GC0 is shown in Figure 1. The portion of the circuit inside the outer line is inside the Apple computer. THe 150 K potentiometer on the outside is a game paddle, which can be replaced by a themistor (for measuring temperature) or a phototransistor (for measuring light or time).

To access the available input and output functions of the game paddle I/O for those of you who will want to build your own probes and sensors, it becomes a matter of knowing which of the pins on either the 16 pin internal connector or the 9 pin external port (on Apple IIe & IIc). A diagram and pin description is presented below.

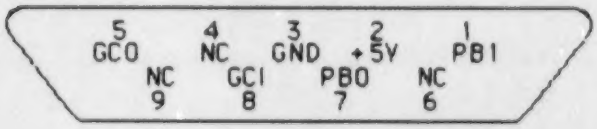
FIGURE 1



Pin assignments for 16 and 9 pin I/O connectors

9	-NC	GND-	8
10	-GC1	GC2-	7
11	-GC3	GC0-	6
12	-AN3	STROBE-	5
13	-AN2	PB2-	4
14	-AN1	PB1-	3
15	-AN0	PB0-	2
16	-NC	+5V-	1

PB = PUSHBUTTON SWITCH
AN = ANNUNCIATOR
GC = GAME CONTROL
NC = NO CONNECTOR



PIN(s)	SYMBOL	FUNCTION AND DESCRIPTION
1	+5V	Five volt power supply. Maximum current is 100 milliamps.
2 - 4	PB0-PB2	Single-bit (Pushbutton) inputs. These are standard 74LS series TTL inputs.
6,7,10,11	GC0-GC3	Game-paddle Inputs. Generally these are connected to +5 V through a resistance.
8	GND	Computer System Electrical Ground.
12 - 15	AN0-AN3	Annunciator outputs. These are standard 74LS series TLL outputs.

Attaching this 150 K potentiometer (Apple game paddle) to a mechanical device which will cause a change in the position of the resistor, and one can plot the period of a pendulum, track a rocket on its flight skyward or measure how far it is to a distant object. By replacing the potentiometer with a thermistor one can now measure changes in temperature, breathing rates or the humidity in a classroom. While replacing the thermistor with a phototransistor or a photoresistor will allow your students to measure the speed of a falling object due to gravity, time the period of a pendulum, the speed of a car moving across air track or check the shutter speed on your camera. Through the use of one probe or four, you can receive digital readouts, plot time versus temperature, or determine average acceleration.

Once the basic concepts of interfacing are understood, the only limiting factor to the experiments which can be done or applications of this technology is our own ability to create.

To date I have found the following suppliers currently provide a source for reasonably priced hardware and software with which one can get started:

CROSS EDUCATIONAL SOFTWARE, 1802 North Trenton St., P.O. Box 1536, Ruston, Louisiana 71270 - Offers 3 programs - HEAT, LIGHT and SOUND. Each contains program disk, materials for making your own probes, instructions for probe construction, program uses and hard copy of all interfacing programs. Cost $60.00 each.

VERNIER SOFTWARE, 2920 S.W. 89th Street, Portland, OR 97225 - Precision timer program contains program disk and documentation on timing programs. Cost is $39.95.

SCI-COMP, 2404 Cherokee St., Adelphi, MD 20783 - Science interfacing diskette ($20.00), interfacing module ($45.00), paddle port extender ($16.00), thermometer probe ($10.00), light sensitive probe ($5.00).

COMPONENTS FOR INTERFACING EQUIPMENT

1. POTENTIOMETER - (i.e. variable resister, generally called a "POT"). The APPLE game paddle input converts a resistance from 0 to 150,000 ohms (generally called "150 K" or 0.1 Meg) to screen numbers of 0 to 255. OHN/ELECTRONICS - Palatine, IL. JAIN056815 4UA 4.20 or Radio Shack 271-338 2.00. (This is only 100 K, but will work). Be sure to get a liner taper pot and not an audio taper pot.

2. THERMISTOR - Since the resistance of a thermistor is a function of temperature, it can be made into a temperature probe. A source of thermistors is FENWAL ELECTRONICS, 63 Fountain Street, Faramingham, MA. (617) 872-8841. FENWAL Part Number = GA45P2. Another source is Midwest Components Inc., P.O. Box 787, 1981 Port City Boulevard, Muskegon, MI 49443. Part Number 9RT1H412.

3. PHOTOTRANSISTOR - The resistance of a phototransistor is a function of the light intensity. Phototransistors can be used for photo gates and for measuring light intensity. FPT-100 (Silicon phototransistor) Radio Shack Part Number 276-130. (.89)

4. 16 PIN HEADER - This is used to connect to the internal 16 pin socket inside the APPLE microcomputer. Radio Shack Part Number 276-1980 (1.69)

5. NEON LAMPS with DROPPING RESISTOR FOR 120 VOLTS AC - Radio Shack Part Number 272-1100A (.79) contains two NE2 NEON LAMPS. Used to make a 120 cycle flasher. You will also need some lamp cord and a wall plug. These last two items are available at any hardware store.

6. TRICOLOR LIGHT EMITTING DIODES (commonly called a "LED"). Used to show that AC current is the same as DC current flowing first one direction and then the other direction. Also shows that red light plus green light gives a yellow light. Radio Shack Part Number 276-035 (1.39)

7. INFRARED HIGH OUTPUT LED EMMITTER-Radio Shack Part Number (276-143) (1.49)
   INFRARED PHOTOTRANSISTOR - Radio Shack Part Number 276-145 (.89). These two components can be used to make a photo gate that responses to infrared light rather than visible light as in #3. This makes the photo gate less sensitive to ambient light.

IMPLEMENTATION OF A K-8 COMPUTER CURRICULUM

By: John Tofano and Debbie Field, Elk Grove Community Consolidated School District #59, Arlington Heights, IL

In Community Consolidated School District 59, students are introduced to using the microcomputer at the kindergarten level and this orientation and utilization of the microcomputer for instructional purposes is continued through the eighth grade. At the elementary level, microcomputers are utilized to provide the students with computer assisted instruction activities over a wide range of educational topics. These microcomputers are located in each school's library learning center; all learning centers are staffed with a full time learning center teacher and a minimum of one full time learning center aide., Students are scheduled into the learning center to work on topics in specific subject areas. The library learning center teacher with the assistance of the classroom teacher will recommend the instructional software programs that are to be utilized by each student. Software programs that provide drill and practice, simulations, tutorial and problem solving activities are available and can be selected in order to individualize the students instruction.

At the junior high school level, computer utilization can be categorized as falling in two basic areas. Computer assisted instruction activities comprise one area, the second area consists of activities more readily defined as computer science instruction.

1. Computer Assisted Instruction
   The library learning center in each junior high school is equipped with a minimum of two complete microcomputer systems. Students are scheduled into the learning center in a manner similar to the system described for the elementary schools. During the 1984-85 school year, a word processing instructional computer lab has been implemented as a pilot project in one of the district's four junior high schools. This lab contains 15 Apple IIe computers, 3 printers, video cassette recorder and a 60" projection T.V. system. A full time computer aide is assigned to the lab which enables subject area teachers to schedule the lab for use with their classes. Each junior high school also has the equivalent of one full time teacher that teaches the programming classes and serves as the computer resource teacher for the school.

2. Computer Science and Application Activities
   Junior high school students receive a minimum of 7 weeks of computer instruction at each grade level during their 3 years in junior high school. Each junior high school has a computer lab containing 18 to 20 microcomputers and 3 to 4 printers. The curriculum guide and materials utilized at each grade level have been developed by District 59 staff members. The specific details regarding topics and materials utilized at each grade are described in the latter part of this report.

6th Grade - Logo - Volume 1

This Logo unit includes a teacher's curriculum manual supported by software and student workbooks. Following are the Logo concepts taught at this grade level. A 3 week unit on keyboarding and word processing is also taught at the 6th grade level.

Logo Objectives for Volume I

Les. 1 - Keyboard, Monitor, Disk Drive, Diskette, CS, Fd, BK, RT, LT
Les. 2 - Heading, CTRL - L, T, and S, wrapping, error messages, SETPC
Les. 3 - prediction, HOME, dimensions of screen
Les. 4 - predictions, more than one command on a line
Les. 5 - SETBG, review
Les. 6 - HT, ST, PU, PD, PE, PX
Les. 7 - geometric shapes, exterior angles
Les. 8 - REPEAT
Les. 9 - Circles, parts of circles (arcs), different sized shapes
Les. 10 - Different sized circles, review
Les. 11 - Using a file disk, CATALOG, LOAD, ERALL, running a procedure
Les. 12 - Editing a procedure, ED, SAVE, ERASEFILE
Les. 13 - Teaching procedures in the editor
Les. 14 - PR, using subprocedures, asking PX for animation
Les. 15 - length, width, radius, filling in circles and rectangles
Les. 16 - Project

7th Grade - Logo - Volume II

This unit covers different Logo concepts than the sixth grade unit, but also contains a teacher's curriculum guide, software and student workbook. A 3 week unit covering word processing and keyboarding activities is also taught. The list of Logo objectives for Volume II are as follows:

Logo Objectives for Volume II

Les. 1 - review of Volume I
Les. 2 - subprocedures, CLEARTEXT, POTS, POPS, PO[ ], ER[ ]
Les. 3 - filling in circles and squares
Les. 4 - TURTLE procedure (given all subprocedures)
Les. 5 - WAGON procedure (given parts, student writes MOVE procedures) SETH
Les. 6 - HOUSE procedure (student writes all subprocedures)
Les. 7 - Inputs
Les. 8 - Recursion
Les. 9 - lists, WAIT, counters, recursion
Les. 10 - Printing words, lists, and numbers; SETCURSOR, dimensions of text screen, FIRST, LAST, BF, BL
Les. 11 - MAKE memory locations, READLIST, SENTENCE
Les. 12 - formatting paragraphs, IF
Les. 13 - Project

8th Grade - 7 Week Computer Curriculum

At this level a curriculum guide has been developed that provides students with an introduction to the BASIC programming language. In addition, the utilization of a word processing program is continued at this level. Listed below are the 17 lessons or objectives taught as part of the BASIC programming unit.

8th Grade Introduction to BASIC

Volume I

Les. 1 - Return Key / Space Bar / Back Arrow / Floppy Disk / Getting Started / Running a Program / CATALOG
Les. 2 - Disk Drive / Prompt Character / Reset Key / Esc Key / Shift Key / 0 and Ø / PR #6

Les. 3 - Immediate Execution / Basic Operations + - * / HOME / PRINT
Les. 4 - Numeric Memory Locations / Quotation Marks / PRINT
Les. 5 - NEW / Syntax Error / Debugging / Colon (:)
Les. 6 - Delayed Execution / Computer Programs / Line Numbers / RUN / LIST / VTAB / HTAB
PROJECT (p. 44)
Les. 7 - Flash / Inverse / Normal / Speed
Les. 8 - SAVE / LOAD / LOCK / UNLOCK / DELETE
Les. 9 - GR / TEXT / COLOR / PLOT / Illegal Quantity
Les. 10 - HLIN / VLIN
PROJECT (p. 75)
Les. 11 - LIST / CTRL "S" / REM / Text Window PROJECT (p. 83)
Les. 12 - FOR NEXT LOOP / Pause
Les. 13 - FOR NEXT LOOP / Background Color / Text Window
PROJECT (p. 99)
Les. 14 - Character String / Dollar Sign ($) / Type Mismatch
Les. 15 - INPUT (Numeric) / INPUT (Character String) / Semi-Colon (;) / Skipping Lines
Les. 16 - IF THEN / Reenter / Flow Charts /
Les. 17 - GET
PROJECT (p. 124)

9th Grade Advanced

This is a full semester computer curriculum developed by a committee of junior high school teachers. Support materials include a full array of student worksheets, software diskettes and monographs. These materials were developed with the assistance of instructional computer teachers from High School District 214. This course was established in order to provide the talented junior high student with the necessary background knowledge so that they would have the opportunity for placement in an advanced high school computer course upon their entry into high school. The scope and sequence for this advance computer unit is as follows:

8th Grade Advanced Computer Course
Scope and Sequence

Unit 1 Introduction to the Computer
Unit 2 Flowcharting and Word Processing
Unit 3 History
Unit 4 Introduction to Programming
Unit 5 Processing Data
Unit 6 Social Issues and Futures
Unit 7 Data Entry
Unit 8 Data Inquiry
Unit 9 Careers
Unit 10 Programming Projects

Unit 11 Final Exam

Summary

The development of K - 8 computer curriculum in the school district has been based on two major objectives; the first objective is that CAI should be integrated into the curriculum at all grade levels; the second objective is that the teaching of programming languages will only take place at the junior high school level.

Therefore, using these major objectives as the base for the computer curriculum, students at the elementary level are introduced to the computer through the learning center or through class demonstrations. The emphasis with computers at this level is on computer assisted instruction.

At the junior high school, however, students come into contact with computers in three ways: (1) through computer assisted instruction activities in the learning center or the new word processing and instructional computer lab, (2) through classroom demonstrations and (3) through specific grade level computer courses. During their 3 years in junior high school, each student will receive a minimum of 21 weeks of detailed computer instruction. In the course of their studies, many aspects of computer literacy will be covered.

CIC--COMPUTERS IN THE CURRICULUM

By: Joanne Troutner, Klondike Junior High, West Lafayette, IN

Seventh grade art students are busy working on the Koala Pad and using the Mouse to finish their computer graphics assignment. Four freshmen are huddled in the far corner working on the interactive fiction assignment given in English class. Three eighth grade math students are working on the remaining computers developing their own mazes for fellow students to use when working with the program "Math Maze." If these activities are not typical of the ones happening in your computer lab or library media center, perhaps you need to take a closer look at integrating the computer into the curriculum.

Far too often we, as educators, forget that the true beauty of microcomputers lies in using them as a teaching tool. And, just as often we fail to help fellow teachers gain the knowledge needed to see how to use the computer as a teaching tool. Many times we hide behind a veil of computerese and make fellow educators feel that they must learn to program in order to use a computer. We all know that is not the case. We simply need to help our colleagues learn to evaluate software and recognize those pieces which will fit effectively into teaching units which many times already exist.

This presentation will focus on helping teachers develop units which use the computer as a teaching tool. The steps needed to plan the unit, exemplary software, and ways to use it in various curriculum areas.

When planning a unit, the teacher must consider the following questions:

1. What role will the computer play? Will it be used for:
   --remedial work
   --enrichment purposes
   --required information on an individual or small group basis
   --demonstration with the entire class?

2. Could this role be as easily done with pencil and paper?

3. Does commercial software exist for the material you want to cover?

4. If commercial software exists, is it of sufficient quality to use?

5. Will students be using the computer individually, in small groups, or as an entire class? How many computers will you need?

6. If the entire class will not be on the computer(s) at one time, what will students do for off-computer activities?

7. Are the computer(s) and software available when you intend to use them?

8. Do you need to schedule a planning meeting with your library media specialist?

EXEMPLARY SOFTWARE FOR CURRICULUM INTEGRATION

THE FACTORY--Explorations in Problem Solving. Sunburst Communications, Inc., 1983. 1 computer disk and teacher's guide. (Backup disk included.) $55.00. Gr. 4 - 12. Apple, Atari, Commodore 64, IBM, TRS-80 Color.

This program allows the student to develop a series of machines which can build a product. Students may then challenge another student to create their product or accept the computer's challenge to create a product.

INDUSTRIAL ARTS

* Have students develop predefined products before creating a wood or metal project.

MATHEMATICS

* Use the program to help teach angles.
* Divide the class into teams. Have teams develop challenges for each other.

MEDIA CENTER

* Use an an independent project.
* Develop a product of the day as a challenge. Limit the amount of time students have to work on it.

SCIENCE

* Have lab partners work together and use the program with the entire class as a problem solving exercise.

CODE QUEST: Problem Solving Skills. Sunburst Communications, Inc., 1983. 1 computer disk and teacher's guide. (Backup disk is included.) $55.00. Gr. 5 - Adult. Apple, TRS-80, Atari.

The student is to guess what the mystery object is. He is provided with a variety of seven types of codes. Hints are given through a help option and a partially completed game may be saved on the disk. A teacher's option allows mystery objects to be entered.

SOCIAL STUDIES

* Use the teacher option to enter current events clues and test student's knowledge.

* Research the history of codes.

* Have students develop a bulletin board of historical figures who have relied on codes.

LANGUAGE ARTS

* Have students develop mystery objects and create clues which are all adjectives, nouns, or some other part of speech.

* Have students write mystery stories which use CODE QUEST to solve the mystery.

SCIENCE

* Use the solving of clues to reinforce the scientific problem solving process.

* Write clues about mystery objects for a particular unit of study.

MATHEMATICS

* Analyze the number of times a letter appears. Graph the results.

* Devise clues about famous mathematicians.

* Invent your own code based on a mathematical scheme.

MEDIA CENTER

* Have a theme for the week and develop mystery objects connected with it. Set up the program for independent student use.

* Hold a CODE QUEST tourney where students challenge each other.

MATH MAZE. DCH Software, 1984. 1 computer disk and teacher's guide. (Backup disk included.) $45.00. Gr. K - 8. Apple, Atari, IBM.

Students go through a maze to reach the correct answer to addition, subtraction, multiplication, and division problems. Various levels of difficulty are included.

MATHEMATICS

* Use this program simply as a drill and practice activity.

* Have the students develop their own mazes and challenge other students.

* Hold a MATH MAZE tourney in your classroom or media center.

DECISION SHOP. Public Domain. Provide disk and return postage.

This disk contains three programs which help teach economic concepts. "Kingdom" allows the student to rule his own kingdom in feudal times. "Star Trader" has the student become a star merchant and travel between planets. "Sell Robots" makes the student the owner of a robot store.

SOCIAL STUDIES

* Review economic principles.

* Use as a decision making exercise with the class divided into groups.

* Have a contest to determine who is the best ruler of Malthusia.

THE WHATSIT CORPORATION. Sunburst Communications, Inc., 1983. 1 computer disk with teacher's guide. (Backup disk included). $55.00. Gr. 6 - 12. Apple, TRS-80.

This program allows students working in groups or individually to manage a company. Decisions on advertising, buying other franchises, taking out loans, hiring people, and etc. need to be made by the student. Game progress may be saved for later use.

SOCIAL STUDIES

* Divide the class into groups and use.

* Have a business manager tourney between classes.

* Use as a tool to help practice decision making.

MATH

* Use to provide practice in math skills.

* Use to sharpen problem solving skills.

PUZZLER. Sunburst Communications, Inc., 1984. 1 computer disk with teacher's guide. (Backup disk included.) $55.00. Gr. 3 - 9. Apple.

Five mystery stories are presented. Students use reading strategies to predict and confirm their solutions. The wonderful aspect of this program is that there may be any number of correct solutions for each story.

LANGUAGE ARTS

* Use to help teach inferences.

* Develop student's ability to make predictions.

* Have a school-wide survey--"What is Sammy?"

INFORMASTER. Grolier Educational Corporation. 1984. 1 computer disk with teacher's guide. (Backup disk included.) $64.50 Gr 4 - 9. Apple.

This is a marvelous data base program. It contains a teaching program which describes vocabulary and the operation of data bases. In addition, a game which involves searching a file on animals is included. Students may develop their own templates and then store files. Each file will hold up to 360 records.

LANGUAGE ARTS

* Have student write and store their own book review.

* Create a file of story starters for essay assignments.

* Have students create their own file of vocabulary words and sentences using them.

* Have students develop their own "Author Hall of Fame".

SOCIAL STUDIES

* Have students create a file of information on a specific time period. Use this file as a test grade.

* Provide students with already created files on topics such as civil rights, the reasons for the revolutionary war, or the colonization of the United States. See what conclusions students can draw from these facts.

* Provide facts about a mythical company of economics students. Have them prepare reports concerning the current status and future progress of this company.

* Have groups of students provide a file of information on a country they create. Give this file to a different group and have them write a synopsis or encyclopedia entry for the country.

* Ask a class to create a current events data base for the year. Pay attention to how you design the template.

The rest is up to you. Be creative!!! Use your imagination!!!

Above all--HAVE FUN with this new teaching tool.

EXCELLENCE IN ADMINISTRATION: PLANNING AND PRESENTING

By: Gary W. Tubb, University of South Florida, Tampa, FL

Electronic spreadsheets are planning/presenting programs available for the Apple, IBM, Pet, and TRS-80 personal microcomputers that may be thought of as an easily modifiable accountant's spreadsheet. The most popular spreadsheet programs are Lotus, VisiCalc, Multiplan, and SuperCalc. Over 400,000 copies of VisiCalc have been sold since 1979. Thus, there are now many more uses for the program than just financial planning. But, since making projections was the original use that the authors Daniel Bricklin, 29, and Robert Frankston, 31, had in mind when Dan was a first-year Harvard Business School student in 1978, that is the approach we shall use in discussing the potential use of spreadsheets in education.

To understand the way in which an electronic spreadsheet functions, we need only to understand the way that we solve most problems--using a calculator, a pencil, and a sheet of paper--three universal tools. In most problems, we have a column of numbers we have to sum, then a percentage to multiply by, and to top it off we usually need to divide by some number. Our three universal tools (calculator, pencil & paper) serve us well until we realize that we:

a. forgot a number,
b. keyed in a wrong number,
c. used the wrong percentage,
d. realize that some jerk will change the numbers on us when we we finally get the answer.

This is where an electronic spreadsheet pays-off in planning, replanning, and presenting data. Imagine a sheet of paper with a column of 16 numbers on lines (rows) 2 to 17. Let's call this column of numbers, column A. To find the sum of the numbers (using Lotus or Visicalc) in column A rows 2 to 17, we simply type @SUM (A2..A17) in location A18. Immediately the spreadsheet calculates the sum of the numbers in locations A2 to A17 and places the sum in location A18 on the screen. If you changed one or more of the numbers in locations A2 to A17, then the spreadsheet will recalculate the sum every time you change a number. The microcomputer is your calculator, the screen is your paper, and the cursor is your pencil.

The advantage of an electronic spreadsheet over a calculator, pencil and paper is that you may save a process of manipulating numbers for later use or alteration. For example, the impact of a teacher benefit upon a salary schedule may be ascertained immediately. Using alternative cohort survival rates in enrollment projections as a basis for assessed valuations is a tedious task by hand, but not with a spreadsheet. If one of the bus barns were to burn down and the Superintendent wanted an assessment of the loss and impact upon the school district for an eminent press conference-a spreadsheet with the appropriate data would be very valuable indeed. These and other examples of the applications of spreadsheet templates are the focus of this paper. But remember, the process of manipulating numbers is the province of the computer. Entering the process is the province of the manager and school administrator. The visible consequences of this process are immediate and easy to modify with an electronic spreadsheet. Clearly, an electronic spreadsheet is a tool which a manager can use to plan.

In our first example, we will look at a spreadsheet as if it were a very simple data base. The first row of the spreadsheet lists the names of the fields of data. Each row thereafter represents a single record of information. The example below lists the buses of a small school district, the original cost and date of purchase.

	A	B	C	D	E	F
1	Org Cost	Equipment	Bus #	Seating	Date Acq	Barn
2	6450.00	GMC 1978	1	66	82	C
3	7420.90	FORD 1972	2	50	72	B
4	9999.00	FORD 1979	3	66	80	B
5	8064.60	IH 1977	4	66	77	C
6	14700.00	FORD 1983	5	66	84	B
7	8500.00	GMC 1979	6	66	82	C
8	6992.42	CHEV 1975	7	66	75	A
9	7420.90	FORD 1971	8	50	71	B
10	15175.00	FORD 1983	9	66	84	B
11	8111.30	CHEV 1971	10	66	71	A
12	15250.00	FORD 1983	11	66	84	B
13	13891.12	CHEV 1979	12	66	79	A
14	6800.00	GMC 1978	13	66	82	C
15	9973.98	IH 1975	14	66	78	C
16	11065.62	FORD 1979	15	66	80	B
17	11065.62	FORD 1979	16	66	80	B

The top row A-F represent the columns of the Lotus or VisiCalc, while the numbers in the left column 1-17 represent the row numbers of both spreadsheets. The criterion and output headers are included on the file BUS1. Using the /DATA commands of Lotus, we will SORT and QUERY the bus database.

Now let us look at the important commands that are available in electronic spreadsheets. First, formulas that use numbers from cells within the template may be placed in other cells of the template. The @SUM(A2..A17) is an example of a sophisticated formula that can be used instead of +A2+A3+A4+ ... etc.

Second, a formula or text may be replicated in many cells by a sequence of simple commands. For example, if cells A2 to A17 represented the cost of 16 buses and we wanted to determine what effect an eight percent increase in replacement cost would have on each bus cost; we could type the formula +A1*1.08 into cell A21, +A2*1.08 into A22, etc. The formulas in A21 to A17 are very similar. Consequently, we could use the relocate (or copy) command.

In our example, we have different buses in three bus barns A, B, and C. Suppose that barn C burned down last night and you are faced with an office-full of reporters. First, sort on column F in ascending order, then enter the following formulas:

In column			resulting in
	A18 enter	@SUM(A2..A4)	$28994.84
	A19	@SUM(A5..A11)	$81031.42
	A20	@SUM(A12..A16)	$39788.58

Copy the formulas into column D to find the sum of the sets on the buses in each bus barn by using the following commands:

First move the cursor to A18, then press
/C
Enter range to copy FROM: A18.A20
Enter range to copy TO: D18.D20

The sum of the seats in each bus barn are displayed in column D as 198 seats in barn A, 430 in Barn B, and 330 in barn C. You may calculate the cost per seat by entering the following formula into Column E:

In cell	E18 enter	+A18/D18
	E19	+A19/D19
	E20	+A20/D20

This data is easily graphed and of a great deal of interest. After graphing the data, you may wish to investigate the transportation depreciation model TRANS1 which follows.

By Gary W. Tubb

Rate 6 yrs 0.15 Year 1984

Rate 7th yr 0.10

Orig cost	(A) Equipment	(B) Bus #	(C) Seating	(D) Date Acq	(E) Cost
$6,450.00	GMC 1978	1	66	1982	6450.00
$7,420.90	FORD 1972	2	50	1972	7420.90
$9,999.00	FORD 1979	3	66	1980	9999.00
$8,064.60	IH 1977	4	66	1977	8064.60
$14,700.00	FORD 1983	5	66	1984	0.00
$8,500.00	GMC 1979	6	66	1982	8500.00
$6,992.42	CHEV 1975	7	66	1975	6992.42
$7,420.90	FORD 1971	8	50	1971	7420.90
$15,175.00	FORD 1983	9	66	1984	0.00
$8,111.30	CHEV 1971	10	66	1971	8111.30
$15,250.00	FORD 1983	11	66	1984	0.00
$13,891.12	CHEV 1979	12	66	1979	13891.12
$6,800.00	GMC 1978	13	66	1982	6800.00
$9,973.98	IH 1975	14	66	1978	9973.98
$11,065.62	FORD 1979	15	66	1980	11065.62
$149,814.84					$104,689.84

Transportation Equipment Schedule, 1983-84
(section 29-5, the school code of Illinois)

(F) Addition	(G) Deletion	(H) Cost	(I) Rate	(J) Acc dep 1	(K) Allow	(L) Delete	(M) Acc Dep N	(N) Balance
0.00	0.00	6450.00	0.15	967.50	967.50	0.00	1935.00	4515.00
0.00	7420.90	0.00	0.00	7420.90	0.00	7420.90	0.00	0.00
0.00	0.00	9999.00	0.15	4499.55	1499.85	0.00	5999.40	3999.60
0.00	0.00	8064.60	0.10	7258.14	806.46	0.00	8064.60	0.00
14700.00	0.00	14700.00	0.15	0.00	2205.00	0.00	2205.00	12495.00
0.00	0.00	8500.00	0.15	1275.00	1275.00	0.00	2550.00	5950.00
0.00	6992.42	0.00	0.00	6992.42	0.00	6992.42	0.00	0.00
0.00	7420.90	0.00	0.00	7420.90	0.00	7420.90	0.00	0.00
15175.00	0.00	15175.00	0.15	0.00	2276.25	0.00	2276.25	12898.75
0.00	8111.30	0.00	0.00	8111.30	0.00	8111.30	0.00	0.00
15250.00	0.00	15250.00	0.15	0.00	2287.50	0.00	2287.50	12962.50
0.00	0.00	13891.12	0.15	8334.67	2083.67	0.00	10418.34	3472.78
0.00	0.00	6800.00	0.15	1020.00	1020.00	0.00	2040.00	4760.00
0.00	0.00	9973.98	0.15	7480.49	1496.10	0.00	8976.58	997.40
0.00	0.00	11065.62	0.15	4979.53	1659.84	0.00	6639.37	4426.25
$45,125.00	$29,945.52	$119,869.32		$65,760.40	$17,577.17	$29,945.52	$53,392.04	$66,477.28

ADVANCED TECHNIQUES OF LOGO PROGRAMMING

By Gary W. Tubb, University of South Florida, Tampa, FL

Perhaps the most fundamentally intellectual aspect of Logo is the ability to use the language to investigate interesting problems in a microworld striped of extraneous variables. All of the Logo procedures in this paper are based on Terrapin Version 1.0. The Euclidean Algorithm for finding the greatest common divisor of two numbers is a perplexing recursive process that often confuses gifted children and not-so-gifted adults. The procedures outlined below demonstrate graphically the Euclidean Algorithm and its relationship to the `Golden Rectangle´.

```
TO RECT :L :S
 REPEAT 2 [FD :L RT 90 FD :S RT 90]
END

TO RMD :L :S
 IF :L < :S THEN FD :L OUTPUT :L STOP
 RECT :S :S
 FD :S
 OUTPUT RMD ( :L - :S ) :S
END

TO GCF :L :S
 RECT :L :S
 MAKE "RM RMD :L :S
 IF :RM = 0 THEN PRINT :S STOP
 RT 90
 GCF :S :RM
END
```

Below are procedures that can be used to investigate the relationship of pitch and duration of musical phrases. In this particular example, "God Save the King" (GSK) is broken into four phrases. In the first phrase, the pitch of the three notes is 1+ 1+ and 2+, while the duration of each note is the same (200). The second phrase has a rest of duration 200 (see GSK1). The interesting aspect of "God Save the King" is that the third and fourth phrases may be constructed from the first two phrases. In particular, the third phrase notes have exactly the same duration as the first phrase except that the pitch of the three notes are increased. The last phrase is similar to the second phrase, except that the order of the pitch of the notes is reversed as well as increased. Gregorian chants have similar patterns of musical relationships.

```
TO GSK1
 MAKE "P1 [1+ 1+ 2+]
 MAKE "D1 [200 200 200]
 MAKE "P2 [12 R 1+ 2+]
 MAKE "D2 [200 200 100 200]
END
```

```
TO GSK
 GSK1
 PLAY :P1 :D1
 PLAY :P2 :D2
 PLAY INC 3 :P1 :D1
 PLAY INC 2 RVS :P2 :D2
END

TO INC :AMT :LST
 IF :LST = [] THEN OUTPUT []
 OUTPUT SE ( INC1 :AMT FIRST :LST ) ( INC :AMT BUTFIRST :LST )
END

TO RVS :LST
 IF :LST = [] THEN OUTPUT []
 OUTPUT SE LAST :LST RVS BUTLAST :LST
END

TO PLAY :PITCHES :DURS
 IF :PITCHES = [] STOP
 PLAY.NOTE ( FIRST :PITCHES ) ( FIRST :DURS )
 PLAY ( BF :PITCHES ) ( BF :DURS )
END

TO .CALL.2 :ADDR :INPUT1 :INPUT2
 .CALL :ADDR :INPUT2 + ( ROUND :INPUT1 ) * 65536
END

TO MAKE.OCTAVE :PITCH :OCTIND :PERIOD
 IF :PITCH = 0 STOP
 MAKE ( WORD "# :PITCH :OCTIND ) :PERIOD
 MAKE.OCTAVE :PITCH - 1 :OCTIND :PERIOD * :N.RATIO
END

TO MAKE.PITCHES :PERIOD
 MAKE.OCTAVE 24 "+ :PERIOD
 MAKE.OCTAVE 24 " :PERIOD * 2
 MAKE.OCTAVE 24 "- :PERIOD * 4
END

TO SETUP
 MAKE "#R 2000
 MAKE "FUDGE 3
 MAKE "N.RATIO 1.05946
 MAKE "BASE.PERIOD 150
 MAKE.PITCHES 20
 DOS [BLOAD MUSIC.BIN]
 MAKE "TONE ( .EXAMINE 43634 ) + 256 * .EXAMINE 43635
END

TO PLAY.NOTE :PERIOD :DURATION
 MAKE "PERIOD THING WORD "# :PERIOD
 .CALL.2 :TONE :PERIOD :DURATION * :BASE.PERIOD / :PERIOD
END
```

```
TO FOO :N :OCT
 PLAY.NOTE WORD :N :OCT 50
 FOO :N + 1 :OCT
END

TO INC1 :AMT :WRD
 MAKE "LL LAST :WRD
 IF :LL = "R THEN OUTPUT :WRD
 IF NUMBER? :LL THEN OUTPUT :WRD + :AMT
 OUTPUT WORD ( :AMT + BUTLAST :WRD ) :LL
END
```

```
MAKE "#24+ (20)
MAKE "#23+ (21.1892)
MAKE "#22+ (22.4491)
MAKE "#21+ (23.7839)
MAKE "#20+ (25.1981)
MAKE "#19+ (26.6963)
MAKE "#18+ (28.2837)
MAKE "#17+ (29.9654)
MAKE "#16+ (31.7471)
MAKE "#15+ (33.6348)
MAKE "#14+ (35.6347)
MAKE "#13+ (37.7535)
MAKE "#12+ (39.9983)
MAKE "#11+ (42.3765)
MAKE "#10+ (44.8962)
MAKE "#9+  (47.5657).....

MAKE "#7-  (213.561)
MAKE "#6-  (226.259)
MAKE "#5-  (239.713)
MAKE "#4-  (253.966)
MAKE "#3-  (269.066)
MAKE "#2-  (285.065)
MAKE "#1-  (302.015)
```

The technical aspects of giving a Logo turtle a sense of touch is briefly outlined below. The version of Logo used is Terrapin 1.0. The machine code program below is entered by booting-up regular DOS 3.3 and typing CALL -151 . Then type each of the six lines of numbers. When you have finished, type 99A0L , press RETURN, and you will see the first part of the machine code program listed below. Check that you have entered in the correct pairs of hexadecimal numbers (enter L again for the rest of the listing). Insert a Logo data disk into drive 1 and type the following:

BSAVE COLLISION.BIN, A$99A0, L$D0

This saves the machine code program for later access by a Logo program.

```
]CALL-151

*99A0.99CF  ENTER THE FOLLOWING

99A0- 86 FD AD 83 C0 60 31 94
99A8- D0 1D E6 FD B1 94 85 FE
99B0- A9 7F 25 30 10 11 18 A5
99B8- C2 29 04 F0 0E B1 94 45
99C0- 1C 25 30 D0 02 E6 FD 51
99C8- 94 91 94 60 27 2D 3E 00

*99A0LL BECOMES THE MACHINE CODE PROGRAM

99A0-   86 FD       STX   $FD
99A2-   AD 83 C0    LDA   $C083
99A5-   60          RTS
99A6-   31 94       AND   ($94),Y
```

```
99A8-   D0 1D       BNE   $99C7
99AA-   E6 FD       INC   $FD
99AC-   B1 94       LDA   ($94),Y
99AE-   85 FE       STA   $FE
99B0-   A9 7F       LDA   #$7F
99B2-   25 30       AND   $30
99B4-   10 11       BPL   $99C7
99B6-   18          CLC
99B7-   A5 C2       LDA   $C2
99B9-   29 04       AND   #$04
99BB-   F0 0E       BEQ   $99CB
99BD-   B1 94       LDA   ($94),Y
99BF-   45 1C       EOR   $1C
99C1-   25 30       AND   $30
99C3-   D0 02       BNE   $99C7
99C5-   E6 FD       INC   $FD
99C7-   51 94       EOR   ($94),Y
99C9-   91 94       STA   ($94),Y
99CB-   60          RTS
99CC-   27          ???
99CD-   2D 3E 00    AND   $003E
```

The following procedures use the sense of touch detected when the turtle or the BOX labeled at MYSHAPE collide with a colored pixel. Begin your Logo programs with SETUP. This loads and connects the machine code program COLLISION.BIN to Terrapin Version 1.0 Logo. The TOUCHING procedure is the heart of the program. The PRIORSENSE procedure initializes the TOUCHING procedure to "NOT TOUCHING" status. The procedures on the next page give our roboturtle radar. The procedure RADARSCAN returns the distance to the object.

```
TO SET
 MAKE "SSIZE 230
 MAKE "USHAPE 228
 MAKE "MYSHAPE 39372
 MAKE "SSENSE 253
 MAKE "SCOLOR 254
 DOS [BLOAD COLLISION.BIN]
 .DEPOSIT 26902 32
 .DEPOSIT 26903 160
 .DEPOSIT 26904 153
 .DEPOSIT 26072 32
 .DEPOSIT 26073 160
 .DEPOSIT 26074 153
 .DEPOSIT 26075 234
END

TO .SHAPE :SH
 .DEPOSIT :USHAPE REMAINDER :SH 256
 .DEPOSIT :USHAPE + 1 QUOTIENT :SH 256
END

TO STAMPSHAPE :SH
 .SHAPE 1 HT .SHAPE :SH
END
```

```
TO RADAROFF
 HT .SHAPE 1
 SETXY :XORG :YORG
 .DEPOSIT :SSIZE 1
 ST .SHAPE 0 PD
 PRIORSENSE
END

TO DISTANCE
 OUTPUT SQRT ( ( XCOR - :XORG ) * ( XCOR - :XORG )
             + ( YCOR - :YORG ) * ( YCOR - :YORG )  )
END

TO RADARON :PROXIMITY
 MAKE "XORG XCOR
 MAKE "YORG YCOR
 .SHAPE 1
 HT PU FD 9
 PRIORSENSE
 WHILE [TOUCHING] [FD 1]
 FD :PROXIMITY
 .SHAPE :MYSHAPE
 PRIORSENSE
END

TO RADARSCAN :PROXIMITY :RANGE :INTENSITY
 RADARON :PROXIMITY
 RADARECHO :PROXIMITY :RANGE :INTENSITY
 MAKE "DISTANCE DISTANCE
 PRINT :DISTANCE
 RADAROFF
END

TO SIZE :N
 HT .DEPOSIT :SSIZE :N ST
 PRIORSENSE
END

TO RADARECHO :PROXIMITY :RANGE :INTENSITY
 IF :PROXIMITY > :RANGE THEN STOP
 MAKE "STEP QUOTIENT :PROXIMITY + :INTENSITY :INTENSITY
 SIZE :STEP FD :STEP
 IF ( TOUCHING ) THEN STOP
 RADARECHO :PROXIMITY + :STEP :RANGE :INTENSITY
END

TO TOUCHING
 IF :PRIORSENSE = .EXAMINE :SSENSE THEN OUTPUT "FALSE
 PRIORSENSE OUTPUT "TRUE
END
```

```
TO WHILE :CONDITION :ACTION
 IF NOT ( RUN :CONDITION ) STOP
 RUN :ACTION
 WHILE :CONDITION :ACTION
END

TO PRIORSENSE
 MAKE "PRIORSENSE .EXAMINE :SSENSE
 MAKE "PRIORCOLOR .EXAMINE :SCOLOR
END

TO MOVE.TILL.TOUCH
 PC 4 PD PRIORSENSE
 WHILE [NOT TOUCHING] [FD 10]
END

TO SETUP
 MAKE "SSIZE 230
 MAKE "USHAPE 228
 MAKE "MYSHAPE 39372
 MAKE "SSENSE 253
 MAKE "SCOLOR 254
 DOS [BLOAD COLLISION.BIN]
 .DEPOSIT 26902 32
 .DEPOSIT 26903 160
 .DEPOSIT 26904 153
 .DEPOSIT 26072 32
 .DEPOSIT 26073 166
 .DEPOSIT 26074 153
 .DEPOSIT 26075 234
 HT FULLSCREEN PU
 READPICT "ACC
 RT 180 SETX110 SETY 45
 REPEAT 6 [PICK]
END

TO MOVERIGHT :UNITS
 SETX XCOR + :UNITS
END

TO RETURN.TO.BIN
 PC 0 PD MOVEBACK :DISTANCE PU
 WHILE [XCOR <100] [MOVERIGHT 10]
END

TO PLACE
 MOVE.TILL.TOUCH
 STAMPSHAPE 0
END

TO MOVELEFT :UNITS
 PU SETX XCOR - :UNITS PD
END
```

229

```
TO FINDPLACE
 WHILE [:DISTANCE < :CALDIST +6] [MOVELEFT 5 MOVE.TILL.TOUCH MAKE "DISTANCE
END

TO MOVEBACK :DISTANCE
 PC 0
 REPEAT ( :DISTANCE - 5 ) / 10 [BK 10]
 SETY 45
END

TO CALIBRATE
 PICK
 MAKE "XORG XCOR
 MAKE "YORG YCOR
 FD 10
 MOVE.TILL.TOUCH
 MAKE "CALDIST DISTANCE
 MAKE "DISTANCE :CALDIST
 MOVEBACK :DISTANCE
 PU
END

TO PICK
 PU HT
 SETXY 110 45 SETH 155
 PD PC 4 REPEAT 3 [FD 13]
 ST PC 0 REPEAT 13 [BK 3]
 PU RT 25
END

TO CLEANPALLET
 HT PU PC 4
 SETXY ( - 15 ) ( - 75 - RANDOM 3 )
 SETH 90 PD
 REPEAT 20 [FD 142 RT 90 FD 1 RT 90 FD 142 LT 90 FD 1 LT 90]
 PC 0 PALLETLOC SETX RANDOM 35
 PD TRIANGLE 18
 PALLETLOC SETX 50 + RANDOM 35
 PD TRIANGLE 18 PU
END

TO TRIANGLE :SZ
 IF :SZ < 1 THEN STOP
 REPEAT 3 [FED :SZ LT 120]
 FD 1 LT 90 FD 1 RT 90
 TRIANGLE :SZ - 2
END

TO PALLETLOC
 PU SETY ( - 75 ) SETH 150
END
```

```
TO CRUDEPALLET
 SETUP
 CLEANPALLET
 CALIBRATE
 FINDPLACE
 PLACE
 RETURN.TO.BIN
 CALIBRATE
 FINDPLACE
 PLACE
 RETURN.TO.BIN
 END
```

ATHLETIC ADMINISTRATION....SPORT STATISTICS

By: Jerry Turry, Niles West High School, Skokie, IL

Today, there are many computer programs written specifically for athletics. They range from those which are inexpensive ($25) to some very sophisticated ones used by professional teams. Each has its strengths and weaknesses. The biggest weakness of any commercial sport statistics is that it is not tailored to the specific needs of each user. In addition, it gets to be prohibitively expensive to spend $50-$100 for each program's statistical needs. On the other hand, commercial packages allow the buyer to begin entering his team data for immediate output. There is little or no need for the user to analyze the sport in order to decide what best suits his needs.

At Niles West, I found it impossible to stay current with the many programs available, and therefore I was unable to recommend a specific purchase to my coaches based on their needs. In addition, they know next to nothing about computers, this makes it unlikely that they would be able to analyze what was available to them and make a wise purchase. Thus, I began to build our own library of `stat' packages using a powerful database, D.B. Master by Stoneware.

There is no doubt that many of the general databases will be adaptable to creating statistical analysis of your sport programs. However, you will really need to have a powerful one if you wish to do a thorough job, since good analysis requires averages, rankings and other complex sorting manipulations of the data.

The fields per file limitation of the database is a critical factor in creating a file for statistics. Unless your database is capable of storing 25 or more fields, you won't have an easy time in file creation. You will be able to create a system, but you'll be significantly limited in your final reports.

The baseball, wrestling and football statistics used in this presentation are self-explanatory. The key is to select the type of fields you'll need. It is critical to involve your coach in this process, since he's the one who is going to be analyzing the data. At Niles West, each of my coaches brings in the raw data from the contest. My student assistant enters it and prints the various reports. My coaches do not really have the time to gather and record the data. In any event I prefer to keep it since it gives me some insight into the programs that I might not witness consistently.

You should strive to create the reports in a manner that is easily reviewed. This means that the reports should be printed with 1 line per record. If the database you select doesn't allow you to clearly identify the column headings, your reports will be difficult, if not impossible to analyze.

An excellent procedure for file creation is to review some of the commercial software to access the fields used for a particular sport. For example, my football statistics are developed around a pre-existing package, with several improvements my coach wanted. He felt it important to record forced fumbles, sacks and first downs. I added these fields. Since football is such a complex game with so many statistics to record, I had to break the file into two components, i.e. offense and defense, otherwise the speed of operating the

program would have been severely limited.

The wrestling program was the most fun to create, since there was nothing on the market commercially to compare. I worked out the details with one of my wrestling coaches. In the illustration that follows, those of you who understand wrestling, will see that just about everything that can happen is included. The file has allowed our coaching staff no only to analyze individual performance, but to set individual and team goals based on this year's data. Prior to the use of the computer, the coaching staff would not have attempted to keep track of all these possibilities.

The baseball statistics were also created with the input from my coaching staff. We followed the format of the scorebook we presently use to maintain individual and team statistics. My coach wanted to know several additional factors, such as percentage of strikes thrown by the pitcher. It was easy to add the necessary fields. The reports generated consist of offensive stats for the batter/runner, defensive stats for the pitches, and fielding statistics. (not shown)

In summary, I would only recommend purchase of a commercial package to those who have no desire to tailor their statistics or be creative. If your program is like mine, you've got some very fine coaches who are creative people and will rise to the challenge. I urge you to attempt file creation tailored to their needs, otherwise they'll never be satisfied. In addition, you'll find it rewarding and quite cost effective. It will not only justify the original purchase price of the database, but you'll be able to do other administrative functions as well. I am available for further consultation. Write to me at Niles West High School, 5701 Oakton Street, Skokie, IL 60077.

INDIVIDUAL SUMMARY
NILES WEST HIGH SCHOOL WRESTLING STATISTICS REPORTING

*****WRESTLER INFORMATION******						*****POINTS SCORED****						****POINTS AGAINST****						****DECISIONS FOR****						**DECISIONS AGAINST*						*RECORD*	
NAME	WT	LV	OPPONENT	SCH	ESC	TAKE DOWN	RVS	NR>2 FALL	NR>3 FALL	ESC	TAKE DOWN	RVS	NR>2 FALL	NR>3 FALL	WINS	MAJR	SPR	FALL	DQ	WINS	MJR	SPR	FALL	DQ	WON	LOSS
WILTGEN	132	V	DAUT	LA	0	1	1	4	1	0	0	1	1	0	0	1	0	0	0	0	0	0	0	0	1	0
WILTGEN	132	V	UNKNOWN	LF	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0
WILTGEN	132	V	GLOUDEMAN	LEY	4	1	0	0	0	2	4	0	0	0	1	0	0	0	0	1	0	0	0	0	0	1
WILTGEN	132	V	CASSATA	GDN	0	3	2	0	0	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	1	0
WILTGEN	132	V	SALLIN	GBS	2	3	0	1	0	1	1	1	0	2	1	0	0	0	0	1	0	0	0	0	0	1
WILTGEN	132	V	BASSUK	GBN	0	1	3	0	0	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	1	0
WILTGEN	132	V	DATLOVSKY	GRT	0	0	0	0	0	0	1	1	2	0	0	0	0	0	0	0	1	0	0	0	0	1
WILTGEN	138	V	IHSA REGNL		0	0	0	0	0	0	0	0	0	0	2	1	0	0	0	2	0	0	0	0	1	2
WILTGEN	138	V	GAGENHIEME	F	3	3	0	0	0	1	2	3	3	1	0	0	0	0	0	0	1	0	0	0	0	1
WILTGEN	138	V	FURLAN	HC	3	3	1	4	1	0	1	2	0	0	0	0	1	0	0	0	0	0	0	0	1	0
----TOTALS FOR WILTGEN----					12	15	7	9	2	6	9	10	6	3	3	2	1	0	0	4	2	0	0	0	6	6
----------GRAND TOTALS----------					278	434	192	217	182	239	288	216	90	58	124	27	34	92	17	107	15	4	43	9	294	178

REG SEASON SUMMARY

NILES WEST HIGH SCHOOL WRESTLING STATISTICS REPORTING

WK: 10
MØ: 1

****POINTS SCORED*****												****POINTS AGAINST****															
*******WRESTLER INFORMATION********						TAKE		NR>2	NR>3			TAKE		NR>2	NR>3	*****DECISIONS FOR***					**DECISIONS AGAINST**					*SCORE*	
NAME	WT	LV	OPPONENT	SCH	ESC	DOWN	RVS	FALL	FALL	ESC	DOWN	RVS	FALL	FALL	WINS	MAJR	SPR	FALL	DQ	WINS	MAJR	SPR	FALL	DQ	NW	OPP
ALBACH	098	V	FORFEIT	HP	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	6	0
BINSTOCK	105	V	PURI	HP	1	3	0	3	3	1	0	0	0	0	0	0	1	0	0	0	0	0	0	0	5	0
UNTERBERGER	112	V	CINQUI	HP	1	4	0	2	2	1	0	1	0	0	0	0	1	0	0	0	0	0	0	0	5	0
LOCHNER	119	V	JOHN DOE	HP	0	4	1	1	0	4	1	0	0	0	0	0	0	1	0	0	0	0	0	0	6	0
BRAVOS	126	V	JAFFE	HP	1	2	0	1	1	1	0	1	0	0	0	1	0	0	0	0	0	0	0	0	4	0
NUTTING	132	V	MORTON	HP	0	0	2	3	0	0	1	1	0	0	0	1	0	0	0	0	0	0	0	0	4	0
GERSHBEIN	138	V	LIU	HP	2	1	0	1	0	1	3	0	0	0	0	0	0	0	0	1	0	0	0	0	0	3
KOLAR	145	V	ROSEN	HP	0	1	1	0	3	0	0	1	0	0	0	1	0	0	0	0	0	0	0	0	4	0
ST. MARIE	155	V	LENS	HP	0	1	1	0	2	0	0	2	0	0	0	0	0	1	0	0	0	0	0	0	6	0
MIRKOVIC	167	V	LEVINSOHN	HP	0	1	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	6	0
RUEHRDANZ	185	V	PALANORI	HP	0	1	1	1	1	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	6	0
MCCULLOH	099	V	SERACI	HP	0	1	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	6	0
------TOTALS FOR 1					5	19	6	12	12	8	5	6	0	0	0	3	2	5	1	1	0	0	0	0	58	3
------TOTALS FOR 10					5	19	6	12	12	8	5	6	0	0	0	3	2	5	1	1	0	0	0	0	58	3

VS SCHURZ

NILES WEST HIGH SCHOOL BASEBALL/SOFTBALL OFFENSIVE REPORT

		*********HITTING**********					********SLUGGING******					******STRIKE OUTS & ON BASE*******								***RUNNING**		
PLAYER	OPPONENT	ABS	RUNS	HITS	RBIS	AVERG	1B	2B	3B	HRS	AVERG	SO	KO	BB	WP	OB	SAC	LOB	PRONT	SBA	SB	PRONT
DAHM	SCHURZ	4	1	2	1	0.500	1	1	0	0	0.750	1	0	1	0	3	0	0	0.750	1	1	1.000
LOCHNER	SCHURZ	5	2	2	0	0.400	2	0	0	0	0.400	0	2	0	0	2	0	2	0.400	1	1	1.000

VS SCHURZ

NILES WEST HIGH SCHOOL BASEBALL/SOFTBALL PITCHING REPORT

PLAYER	OPPONENT	W	L	SV	INNING	ABS	RUNS	HITS	SO	BB	ER	AVERG	1B	2B	3B	HR	E	LOB	HP	WP	BK	1B	BALL	STRK	PTCHS	PRONT
DAHM	SCHURZ	0	0	1	0.33	0	0	0	0	0	0	0.00	0	0	0	0	0	0	0	0	0	0	4	17	21	0.810
KLANDNIK	SCHURZ	1	0	0	5.33	30	2	2	7	4	2	2.63	2	0	0	0	1	5	0	1	0	0	45	71	116	0.610

INDIVIDUAL PLAYER SUMMARY
NILES WEST HIGH SCHOOL DEFENSIVE REPORT

		TACKLING								NILES WEST KICKING GAME									
PLAYER	OPPONENT	INT	FR	FF	SK	UAT	ASST	1-DN	DP	#KR	YRDS	AVERG	#PR	YRDS	AVERG	#FN	YDS	AVERG	#KO	YDS	AVERG
PANCHESIN	GLENBROOK SO	0	0	0	0	2	2	0	0	0	0	0.00	0	0	0.00	0	0	0.00	0	0	0.00
PANCHESIN	HIGHLAND PARK	2	0	0	0	6	6	0	0	0	0	0.00	0	0	0.00	0	0	0.00	0	0	0.00
PANCHESIN	DEERFIELD	0	0	0	0	1	7	0	0	0	0	0.00	0	0	0.00	0	0	0.00	0	0	0.00
----TOTALS FOR PANCHESIN		2	1	0	0	15	34	0	0	0	0		0	0		0	0		0	0	
PAUSTIAN	MAINE WEST	0	0	0	0	1	0	0	0	0	0	0.00	0	0	0.00	0	0	0.00	0	0	0.00
----TOTALS FOR PAUSTIAN		0	0	0	0	1	0	0	0	0	0		0	0		0	0		0	0	
POMERANTZ	FOREST VIEW	0	0	0	0	2	0	0	0	0	0	0.00	0	0	0.00	0	0	0.00	0	0	0.00
POMERANTZ	MAINE WEST	0	0	0	0	0	2	0	0	0	0	0.00	0	0	0.00	0	0	0.00	0	0	0.00
POMERANTZ	GLENBROOK NO	0	0	0	0	0	9	0	0	0	0	0.00	0	0	0.00	0	0	0.00	0	0	0.00
POMERANTZ	WAUKEGAN WEST	0	0	0	0	0	1	0	0	0	0	0.00	0	0	0.00	0	0	0.00	0	0	0.00
POMERANTZ	GLENBROOK SO	0	0	0	0	1	1	0	0	0	0	0.00	0	0	0.00	0	0	0.00	0	0	0.00
POMERANTZ	HIGHLAND PARK	0	0	0	0	0	3	0	0	0	0	0.00	0	0	0.00	0	0	0.00	0	0	0.00
POMERANTZ	DEERFIELD	0	1	0	0	1	6	0	0	0	0	0.00	0	0	0.00	0	0	0.00	0	0	0.00
----TOTALS FOR POMERANTZ		0	1	0	0	4	22	0	0	0	0		0	0		0	0		0	0	
PULLANO	FOREST VIEW	0	0	0	0	6	9	0	0	0	0	0.00	0	0	0.00	0	0	0.00	0	0	0.00
PULLANO	MAINE WEST	0	0	1	1	7	5	0	0	0	0	0.00	0	0	0.00	0	0	0.00	0	0	0.00
PULLANO	MAINE SOUTH	0	0	0	0	7	4	0	0	0	0	0.00	0	0	0.00	0	0	0.00	0	0	0.00
PULLANO	GLENBROOK NO	0	0	0	0	4	4	0	0	0	0	0.00	0	0	0.00	0	0	0.00	0	0	0.00
PULLANO	WAUKEGAN WEST	1	0	0	0	4	5	0	0	0	0	0.00	0	0	0.00	0	0	0.00	0	0	0.00
PULLANO	GLENBROOK SO	0	0	0	0	4	4	0	0	0	0	0.00	0	0	0.00	0	0	0.00	0	0	0.00
PULLANO	HIGHLAND PARK	0	0	0	0	5	8	0	0	0	0	0.00	0	0	0.00	0	0	0.00	0	0	0.00
PULLANO	DEERFIELD	0	0	0	1	4	4	0	0	0	0	0.00	0	0	0.00	0	0	0.00	0	0	0.00
----TOTALS FOR PULLANO		1	0	1	2	41	43	0	0	0	0		0	0		0	0		0	0	
ST. MARIE	GLENBROOK NO	0	0	0	0	0	1	0	0	0	0	0.00	0	0	0.00	0	0	0.00	0	0	0.00
ST. MARIE	WAUKEGAN WEST	0	0	0	0	2	0	0	0	0	0	0.00	0	0	0.00	0	0	0.00	0	0	0.00
ST. MARIE	GLENBROOK SO	0	0	0	0	0	2	0	0	0	0	0.00	0	0	0.00	0	0	0.00	0	0	0.00
----TOTALS FOR ST. MARIE		0	0	0	0	2	3	0	0	0	0		0	0		0	0		0	0	
YOO	FOREST VIEW	0	0	0	0	0	1	0	0	0	0	0.00	0	0	0.00	0	0	0.00	0	0	0.00

INDIVIDUAL PLAYER SUMMARY
NILES WEST HIGH SCHOOL FOOTBALL OFFENSIVE REPORT

		RUSHING					PASSING							RECEIVING				SUMMARY					
PLAYER	OPPONENT	CAR	PM	YDS	AVERG	TD	ATT	CMP	YDS	AVERG	PRONT	TD	INT	RCP	YDS	AVERG	TD	TOT YDS	1/DN	PAT	FG	DP	POINTS
FORNARIS	MAINE SOUTH	11	0	34	3.09	0	0	0	0	0.00	0.000	0	0	0	0	0.00	0	34	0	0	0	0	0
FORNARIS	GLENBROOK NO	9	1	71	7.89	0	0	0	0	0.00	0.000	0	0	0	0	0.00	0	71	0	0	0	0	0
FORNARIS	WAUKEGAN WEST	18	1	76	4.22	0	0	0	0	0.00	0.000	0	0	0	0	0.00	0	76	0	0	0	0	0
FORNARIS	GLENBROOK SO	6	1	34	5.67	0	0	0	0	0.00	0.000	0	0	1	-3	-3.00	0	31	0	0	0	0	0
FORNARIS	HIGHLAND PARK	19	0	90	4.74	0	0	0	0	0.00	0.000	0	0	1	17	17.00	0	107	0	0	0	0	0
FORNARIS	DEERFIELD	10	0	31	3.10	1	0	0	0	0.00	0.000	0	0	1	-3	-3.00	0	28	0	0	0	0	6
T O T A L S FOR FORNARIS		90	3	452		2	0	0	0			0	0	3	11		0	463	0	0	0	0	12
HENNESSY	WAUKEGAN WEST	1	0	1	1.00	0	0	0	0	0.00	0.000	0	0	0	0	0.00	0	1	0	0	0	0	0
HENNESSY	GLENBROOK SO	1	0	-7	-7.00	0	0	0	0	0.00	0.000	0	0	0	0	0.00	0	-7	0	0	0	0	0
T O T A L S FOR HENNESSY		2	0	-6		0	0	0	0			0	0	0	0		0	-6	0	0	0	0	0
KASPRAK	HIGHLAND PARK	2	0	7	3.50	0	0	0	0	0.00	0.000	0	0	2	13	6.50	0	20	0	0	0	0	0
KASPRAK	DEERFIELD	5	0	28	5.60	0	0	0	0	0.00	0.000	0	0	0	0	0.00	0	28	0	0	0	0	0
T O T A L S FOR KASPRAK		7	0	35		0	0	0	0			0	0	2	13		0	48	0	0	0	0	0
KING	FOREST VIEW	2	0	7	3.50	0	0	0	0	0.00	0.000	0	0	1	6	6.00	0	13	0	0	0	0	0
KING	MAINE WEST	8	0	18	2.25	1	0	0	0	0.00	0.000	0	0	0	0	0.00	0	18	0	0	0	0	6
KING	MAINE SOUTH	8	0	34	4.25	0	0	0	0	0.00	0.000	0	0	0	0	0.00	0	34	0	0	0	0	0
KING	GLENBROOK NO	7	0	29	4.14	1	1	0	0	0.00	0.000	0	0	1	2	2.00	0	31	0	0	0	0	6
KING	WAUKEGAN WEST	18	0	90	5.00	2	0	0	0	0.00	0.000	0	0	0	0	0.00	0	90	0	0	0	0	12
KING	GLENBROOK SO	7	0	93	13.29	1	0	0	0	0.00	0.000	0	0	0	0	0.00	0	93	0	0	0	0	6
KING	HIGHLAND PARK	7	0	14	2.00	1	0	0	0	0.00	0.000	0	0	1	8	8.00	0	22	0	0	0	0	6
KING	DEERFIELD	12	0	41	3.42	1	0	0	0	0.00	0.000	0	0	0	0	0.00	0	41	0	0	0	0	6
T O T A L S FOR KING		69	0	326		7	1	0	0			0	0	3	16		0	342	0	0	0	0	42
KLANDNIK	FOREST VIEW	6	0	27	4.50	0	0	0	0	0.00	0.000	0	0	3	35	11.67	0	62	0	0	0	0	0
KLANDNIK	MAINE WEST	14	0	69	4.93	0	0	0	0	0.00	0.000	0	0	2	21	10.50	0	90	0	0	0	0	0
KLANDNIK	MAINE SOUTH	11	0	69	6.27	1	0	0	0	0.00	0.000	0	0	5	108	21.60	1	177	0	0	0	0	12
KLANDNIK	GLENBROOK NO	4	0	13	3.25	0	0	0	0	0.00	0.000	0	0	0	0	0.00	0	13	0	0	0	0	0
KLANDNIK	WAUKEGAN WEST	10	0	47	4.70	0	0	0	0	0.00	0.000	0	0	1	20	20.00	0	67	0	1	0	0	1
KLANDNIK	GLENBROOK SO	11	0	53	4.82	0	0	0	0	0.00	0.000	0	0	0	0	0.00	0	53	0	0	0	0	0
T O T A L S FOR KLANDNIK		56	0	278		1	0	0	0			0	0	11	184		1	462	0	1	0	0	13

TEAM SUMMARY
NILES WEST HIGH SCHOOL FOOTBALL OFFENSIVE REPORT

PLAYER	OPPONENT	RUSHING CAR	PM	YDS	AVERG	TD	PASSING ATT	CMP	YDS	AVERG	PRONT	TD	INT	RECEIVING RCP	YDS	AVERG	TD	SUMMARY TO. YDS	1/DN	PAT	FG	DP	POINTS
NILES WEST	FOREST VIEW	26	0	111	4.27	0	21	6	58	9.67	0.286	0	1	6	58	9.67	0	169	5	0	0	0	0
NILES WEST	MAINE WEST	49	1	284	5.80	5	8	3	51	17.00	0.375	0	0	3	51	17.00	0	335	13	5	0	0	35
NILES WEST	MAINE SOUTH	35	2	161	4.60	1	18	10	226	22.60	0.556	2	2	9	226	25.11	2	387	13	0	0	0	18
NILES WEST	GLENBROOK NO	29	3	108	3.72	1	6	1	2	2.00	0.167	0	1	1	2	2.00	0	110	9	0	0	0	6
NILES WEST	WAUKEGAN WEST	52	3	225	4.33	2	12	2	26	13.00	0.167	0	4	2	26	13.00	0	251	16	1	0	0	13
NILES WEST	GLENBROOK SO	32	3	148	4.63	1	9	2	11	5.50	0.222	0	0	0	0	0.00	0	148	7	0	0	0	6
NILES WEST	WAUKEGAN E> FORFT	0	0	0	0.00	0	0	0	0	0.00	0.000	0	0	0	0	0.00	0	0	0	0	0	0	0
NILES WEST	HIGHLAND PARK	36	0	83	2.31	1	11	8	94	11.75	0.727	0	0	8	94	11.75	0	177	15	1	1	0	18
NILES WEST	DEERFIELD	39	0	139	3.56	2	15	3	22	7.33	0.200	0	4	3	22	7.33	0	161	12	1	0	0	13
	GRAND TOTALS	298	12	1259		13	100	35	490			2	12	32	479		2	1738	90	8	1	0	101

TEAM SUMMARY
NILES WEST HIGH SCHOOL FOOTBALL DEFENSIVE REPORT

PLAYER	OPPONENT	TACKLING INT	FR	FF	SK	UAT	ASST	OPPONENT'S OFFENSE & KICKING GAME 1-DN	PTS	#KR	YDS	AVERG	#PR	YDS	AVERG	#PN	YDS	AVERG	#KO	YDS	AVERG
NILES WEST	FOREST VIEW	0	0	0	0	25	36	10	22	3	32	10.67	0	0	0.00	5	124	24.80	0	0	0.00
NILES WEST	MAINE WEST	1	3	1	2	32	28	7	12	2	24	12.00	1	0	0.00	2	68	34.00	6	191	31.83
NILES WEST	MAINE SOUTH	0	1	0	0	22	30	9	19	3	27	9.00	3	28	9.33	4	145	36.25	4	127	31.75
NILES WEST	GLENBROOK NO	0	0	0	1	22	49	14	19	2	71	35.50	5	62	12.40	3	105	35.00	4	180	45.00
NILES WEST	WAUKEGAN WEST	2	2	1	1	19	28	4	10	2	18	9.00	2	28	14.00	5	174	34.80	3	150	50.00
NILES WEST	GLENBROOK SO	1	1	0	0	23	26	17	38	3	22	7.33	4	46	11.50	1	32	32.00	7	255	36.43
NILES WEST	WAUK E> FORFT	0	0	0	0	0	0	0	0	0	0	0.00	0	0	0.00	0	0	0.00	0	0	0.00
NILES WEST	HIGHLAND PARK	2	0	0	0	29	38	13	12	2	20	10.00	3	12	4.00	3	105	35.00	3	132	44.00
NILES WEST	DEERFIELD	0	1	0	3	27	41	15	37	2	37	18.50	0	0	0.00	2	80	40.00	7	355	50.71
	GRAND TOTALS	6	8	2	7	199	276	89	169	19	251		18	176		25	833		34	1390	

INDEX

A
Achberger, Fred, 19
ACM/IEEE, 19
Adams, David R., 4
Administration, 184, 205, 217
AFIPS, 1, 77
Aiken, Robert M., 19
Alexander, Wilma Jean, 4
Anderson, Eric S., 59
Arch, John C., 19
Athletics, 217

B
Badrkhan, Kamiran, 4
BASIC, 72, 82
Biology, 79
Bortnick, Robert, 62
Burton, Paul, 66

C
CAI, 153
Career education, 95
Carmony, Lowell, 40, 72
Charp, Sylvia, 77
Communication Skills, 134
Computer engineering, 4
Computer literacy, 4, 99, 173
Computer science, 4
Counseling, 184
Crovello, Theodore J., 79
Cummins, Jerry, 40, 82
Curricular, 62, 79, 82, 95, 97, 100, 142
Curriculum
  Elementary, 103, 116, 127, 145, 148, 196
  Junior High, 193, 200
  Secondary, 1, 4, 19, 40, 99, 110, 122, 136
  Vocational, 113

D
Daggett, Willard, 4
Dana, Ann, 86
David, Andrew, 90
Dennis, Richard, 95
Doyle, James M., 4

E
Elementary, 97, 127, 142, 145, 196
Engebretson, Mary Ann, 97
Enns, Don, 40

F
Field, Debbie, 196
Finegan, Patricia, 136

G
Gjertsen, Annie, 99
Goodson, Bobby, 100
Gordon, Michael, 103
Gruenkemeyer, Greg, 40

H
Haney, Michael R., 19
Hutinger, Patricia L., 107

I
IBM PCjr, 205
ICTM, 40
Ireland, Jeannie, 110, 113

J
Jackson, Truman, 116
Junior High, 191, 193, 200

K
Kelly, Bernard, 40, 118
Keyboarding, 97, 116, 127, 145
Klemm, Hildegard A., 4

L
Laird, Bruce A., 122
Lawson Jr, John D., 19
Lazzara, Marlene, 127
Lemke, Cheryl, 19
Library, 59
Lloyd, Gary M., 4
Logo, 90
Logo Advanced, 209
Luetke-Stahlmann, Barbara, 131
Luetke-Stahlmann, Kent, 131
Lykos, Peter, 1, 134

M
Macintosh Computer, 72, 140
McConnell, John, 40, 136
Medley, Donald B., 4
Messersmith, Bill, 40
Moll, Robert, 140
Mueller, Karen L., 142
Mueller, Mark, 145

N
Nelson, Richard F., 40

O
O'Brien, Thomas C., 148, 153

P
Parker, Judy, 127
Pascal, 140
Pasewark, William, 4
Peripherals, 191
PFS Series, 184
Poage, Julie A., 161
Poland, Robert, 4
Porter, Billie, 40
Pretty, Cecil, 163
Problem solving, 136, 153
Programming, 4, 19, 72, 82, 90, 140, 191, 209

Q
Quick, Carolyn, 170

R
Research, 161
Rine, David, 173
Robotics, 4
Rogers, Jean B., 19

S
Schaljo, Raymond, 40
Science, 193
Secondary, 1, 4, 19, 40, 72, 79, 82, 95, 99, 110, 122, 136, 140, 217
Shattuck, Penny, 177
Simmons, Jane A., 184
Software evaluation, 100
Special Education, 107, 131, 170, 177
Sullons, Ralph, 191
Swanson, Thomas A., 19

T
Tarnow, Frederic, 193
Technology, 118, 163
Telecommunications, 66
Tofano, John, 196
Troutner, Joanne, 200
Tubb, Gary W., 205, 209
Tumolo, Samuel F., 19
Turry, Jerry, 217

V
Van Hook, Victor, 4
Vocational Education, 113

W
Word Processing, 62, 86, 127, 142

VENDORS

Ted Anspaugh
ADDISON-WESLEY PUBLISHING
1843 Hicks Road
Rolling Meadows, IL 60008

Gayle Aylsworth
APPLE COMPUTER INC.
8700 W. Bryn Mawr
Chicago, IL 60631

Donald D. Lacy
APPLIED COMPUTER SYSTEMS
3033 Reynoldsburg/New Albany Road
New Albany, OH 43054

Neal R. Dorsey
BECKLEY-CARDY
1645 Downs Drive
West Chicago, IL 60185

Gerald Berkowitz
BURK SOFTWARE
P.O. Box 515
Park Ridge, IL 60068

John Carr
COMARK INC.
481 W. Fullerton
Elmhurst, IL 60126

Richard Dlugo
COMP-UNIQUE
4615 Clausen Avenue
Western Springs, IL 60558

Mary Wallace
COMPUTER CURRICULUM CORPORATION
P.O. Box 10080
Palo Alto, CA 94303

Robert W. Cook
COMPUTER RESOURCES, INC.
Barrington Mall
Barrington, NH 03825

Barbara Durlacher
CORONADO PUBLISHERS
1250 Sixth Avenue
San Diego, CA 92101

Steven Murdock
CURRICULUM ASSOCIATES
5 Esquire Road
North Billerica, MA 08162

Kathleen Gedemer
DLM TEACHING RESOURCES
1105 Wadsworth Place
Vernon Hills, IL 60061

Joel Reich
ECOTRAN SYSTEMS
21111 Chagrin Blvd.
Beachwood, OH 44122

Barbara Neu
EDUCATIONAL ACTIVITIES, INC.
1937 Grand Avenue
Baldwin, NY 11510

Ron Samuels
EDUCATIONAL COMPUTER SYSTEM
ACORN COMPUTER
870 E. Higgins
Schaumburg, IL 60195

Maizie Patterson
ENCYCLOPAEDIA BRITTANICA
EDUCATIONAL CORPORATION
425 N. Michigan Avenue
Chicago, IL 60611

Denise Boucher
FOLLETT QUALITY COURSEWARE
4506 Northwest Highway
Crystal Lake, IL 60014

Bob Fowler
HARTLEY COURSEWARE, INC.
2023 Aspen Glade
Kingwood, TX 77339

Jim Barr
IBM
One Continental Towers
1701 Golf Road
Rolling Meadows, IL 60008

Robert E. Bittner
INDIAN VALLEY COMPUTER
R.R. #1 - Box 243
Tiskilwa, IL 61368

241

Barbara Schultz
JOSTEN LEARNING SYSTEMS, INC.
600 West University Drive
Arlington Heights, IL 60004-1889

Tim Klebe
K-LOG COMPANY
41518 N. Racine
Zion, IL 60199

LeRoy Peterson
LEROY PETERSON AND ASSOCIATES
15508 McKinzie Boulevard
Minnetonka, MN 55345

Fay Paras
MIDWEST VISUAL EQUIPMENT CO.
6500 N. Hamlin Avenue
Chicago, IL 60645

Tom Hoglund
MITCHELL PUBLISHING, INC.
915 River Street
Santa Cruz, CA 95060

Nabih Mangoubi
NABIH'S INC.
515 Davis Street
Evanston, IL 60201

J. Patrick Hogan
NATIONAL COMPUTER SYSTEMS, INC.
Suite 101
1834 Walden Office Square
Schaumburg, IL 60195

Norm Olson
NATIONAL SCHOOL SERVICES
250 North Wolf Road
Wheeling, IL 60090

Patrick D. McCarthy
NEOTEK SOFTWARE, INC.
60 Monee Road
Park Forest, IL 60466

Neal G. Gunderson
NGG ASSOCIATES INC.
P.O. Box 27517
Milwaukee, WI 53277

Catherine Schmidt
PC LEARNING SYSTEMS
7754 Harwood Avenue
Milwaukee, WI 53213

Donna Comber
RADIO SHACK
1340 Remmington Road
Schaumburg, IL 60195

Paul B. Harris
SCHOLASTIC INC.
730 Broadway
New York, NY 10003

Robert J. Fedalis
SOFTWARE CITY
2304 E. Rand Road
Arlington Heights, IL 60004

Ed Kozel
SOUTH-WESTERN PUBLISHING CO.
355 Conde Street
West Chicago, IL 60185

Daniel E. Meyer
SOUTHWEST TECHNICAL PRODUCTS CORP.
219 W. Rhapsody
San Antonio, TX 78216

Hal Olson
SRA
1840 Ashley Road
Hoffman Estates, IL 60195

Rita Smaila
SUNBURST COMMUNICATIONS, INC.
39 Washington Ave.
Pleasantville, NY 10570

Paula Krecier
SVE
1345 W. Diversey Parkway
Chicago, IL 60076

Carol Gazda
UNIVERSAL COMPUTERS
1890 First Street
Highland Park, IL 60035

Terry Shevelenko
VIDEO ETC.
405 Lake Cook Plaza
Deerfield, IL 60015

Sam Geati
VIDEO IMAGES
890 Cambridge Drive
Elk Grove Village, IL 60007

Richard Cloonan
WINNEBAGO SOFTWARE CO.
115 West Main Street
P.O. Box 411
Caledonia, MN 55921

Rhonda Kronfeldt
WM. C. BROWN PUBLISHERS
2460 Kerper Blvd.
Dubuque, IA 52001

243 THRU 247

SPEAKERS

Steve Allen
Allen Communications
140 LakeSide Plaza II
5225 Wiley Post Way
Salt Lake City UT 84116

Marla Altschuler
Scholastic Publishing
385 Sherman Ave.
Suite 1
Palo Alto CA 94306

Arlene Ament
Stanley Field School
2055 Landwehr Rd.
Northbrook IL 60062

Eric Anderson
Micro Computer Libraries
145 Marcia
Freeport IL 61032

Polly Andrews
Highland Junior High School
310 W. Rockland Rd.
Libertyville IL 60048

Phyllis Bielgard
Glenbrook South HS
4000 W. Lake Ave.
Glenview IL 60025

Robert Bortnick
Elk Grove CC Dist. #59
2123 S. Arlington Hts. Rd.
Arlington Heights IL 60005

Paul Burton
Illinois Heartland Computer Consortium
2240 East Geddes
Decatur IL 62526

Lowell Carmony
Lake Forest College
Computer Science Department
Lake Forest IL 60045

Maurice Champagne
Buffalo Grove High School
1100 W. Dundee Road
Buffalo Grove IL 60090

Sylvia Charp
T.H.E. Journal
39 Maple Ave.
Upper Darby PA 19082

Robert W. Cook
Computer Resources, Inc.
Barrington Mall
Barrington NH 03825

Theodore Crovello
Notre Dame University
Department of Biology
University of Notre Dame
Notre Dame IN 46556

Jerry Cummins
Proviso West High School
Wolf Rd. and Harrison St.
Hillside IL 60162

Willard Dagget
Division of Occupational Ed.
New York State Educational Dept.
Room 1623
Albany NY 12224

Ann Dana
Hinsdale District #181
Hinsdale Junior High
100 So. Garfield
Hinsdale IL 60521

Henry R. Datelle
Computer Systems Research, Inc.
Avon Park South
P.O. Box 45
Avon CT 06001

Andy David
5430 S. Harper
Chicago IL 60615

Richard Dennis
University of Illinois
College of Secondary Education
1310 South Sixth Ave. Rm. 398
Champaign IL 61820

Verl Dennis
Arthur Anderson
1405 N. 5th Ave.
St. Charles IL 60174

SPEAKERS

Mary Ann Engebretson
Stillwater Public Schools
Stillwater MN 55082

Don Enns
Highland Park HS
433 Vine
Highland Park IL 60035

Bill Fanning
Acorn Computer Corporation
8608 W. Carmen
Norridge IL 60656

Debbie Field
Elk Grove CC Dist. #59
2123 S. Arlington Heights Rd.
Arlington Heights IL 60005

Patricia Finegan
Glenbrook South HS
4000 W. Lake Ave.
Glenview IL 60025

Leroy Finkel
San Mateo County Office of Education
333 Main Street
Redwood City CA 94063

Annie Gjertsen
Buffalo Grove High School
1101 W. Dundee Road
Buffalo Grove IL 60090

Bruce Gjertsen
MICRO-IDEAS
2701 Central Road
Glenview IL 61832

Bobby Goodson
Computer Using Educators
1666 Kitchener Drive
Sunnyvale CA 94087

Jeffry S. Gordon
Computer Based Instructional Systems
University of Cincinatti
Cincinnati OH 45221-0002

Michael Gordon
Director, CADRE
570 Pleasant Ave.
Glen Ellyn IL 60137

Harry C. Hadland
Classroom Consortia Media, Inc.
57 Bay Street
Staten Island NY 10301

Stephen Halpern
Classroom Consortia Media, Inc.
57 Bay St.
Staten Island NY 10301

Carol Hamlett
Peabody College
Box 328
Vanderbilt University
Nashville TN 37203

Ted Hasselbring
Peabody College
Box 328
Vanderbilt University
Nashville TN 37203

Dan Hicks
IBM
2109 Larkdale
Glenview IL 60025

Kay Hicks
MICRO IDEAS
Springman School
2701 Central
Glenview IL 60025

Patty Hutinger
Western Illinois University
Horrabin Hall Room 27
Macomb IL 61455

Jeannie Ireland
Illinois State University
Department of Home Economics
Normal IL 61761

Truman Jackson
State Specialist for Business Ed.
1510 14th Ave. NW
New Brighton MN 55112

Barbara Johnson
National College of Education
2840 Sheridan Road
Evanston IL 60201

11779

SPEAKERS

Bernie Kelly
High School Dist. #214
Forest View High School
2121 S. Goebbert Rd.
Arlington Heights IL 60005

Peggy Lahs
Abracadabra, Inc.
Mendham NJ 07945

Bruce Laird
Township High School District 211
1750 Roselle Road
Palatine IL 60067

Dennis Lauro
Think Technologies, Inc.
420 Bedford St.
Lexington MA 02137

Marlene Lazarra
Glenview School District# 34
Henking School
2941 Linneman St.
Glenview IL 60025

Barbara Luetke-Stahlman
Northern Illinois University
Gram 227
Dekalb IL 60115

Peter Lykos
Illinois Institute of Technology
Chicago IL 60616

Cheryl Marty-White
Minnesota Educational Computing Corp
(MECC)
3490 Lexington Ave. North
St. Paul MN 55112

Bodie Marx
Mindscape, Inc.
3444 Dundee Rd.
Northbrook IL 60062

Michael M. Mazen
Computer Systems Resource, Inc.
Avon Park South
P.O. Box 45
Avon CT 06001

John McConnell
Glenbrook South High School
4000 W. Lake Ave.
Glenview IL 60025

Robert Moll
132 High Street
Amherst MA 01002

Karen Mueller
Libertyville District #70
Butterfield School
1441 W. Lake Ave.
Libertyville IL 60048

Mark Mueller
Highland Junior High School
310 W. Rockland Road
Libertyville IL 60048

Mark Myers
Mid-Central IL Computer Alliance
LakeLand Community College
South Route 45
Mattoon IL 61938

Robert Naughton
New Trier High School, Athletic Dir
385 Winnetka Ave.
Winnetka IL 60093

Richard Nelson
MICRO-IDEAS
Springman School
2701 Central Rd.
Glenview IL 60025

Thomas C. O'Brien
Southern Illinois University
Teachers Center
Edwardsville IL 62025

Judy Parker
Glenview School District# 34
Henking School
2941 Linneman St.
Glenview IL 60025

Art Peters
Robotic Systems
401 N. Salem

SPEAKERS

Marlene Peterson
Glenbrook South HS
4000 W. Lake Ave.
Glenview IL 60025

Julie Poage
Roycemore School
640 Lincoln St.
Evanston IL 60201

Tom Powers
Glenview School District #34
Henking School
2941 Linneman St.
Glenview IL 60025

Cecil Pretty
MICRO-IDEAS
Springman School
2701 Central Rd.
Glenview IL 60025

Tom Prosen
Sunburst Communications
Room BD9
39 Washington Ave.
Pleasantville NY 10570

Carolyn Quick
Illinois State Board of Education
100 N. State St.
Springfield IL 62777

Don Rausch
Schaumburg District #54
720 Central Road
Matteson IL 60443

Rodney Roberts
IBM Corporation - 4570
P.O. Box 1328
Boca Raton FL 33432

Jean Rogers
University of Texas
Department of Computer Science
Painter 3.28
Austin TX 78712

Raymond Schaljo
Illinois State Board of Education
100 No. First St.
Springfield IL 62777

Mel Schmidt
Niles West High School
Oakton Street at Edens Highway
Skokie IL 60077

Mary Schneider
National College of Education
2840 Sheridan Road
Evanston IL 60201

Michael Schuffert
Fenton High School
1000 W. Green
Bensenville IL 60106

Penny Shattuck
MICRO-IDEAS
2710 Central Road
Glenview IL 60025

Jane Simmons
Maine Township High School Dist. #207
Maine East HS
2601 W. Dempster
Park Ridge IL 60068

Romualdus Skvarcius
Boston University
755 Commonwealth Avenue
Boston MA 02215

Lorence H. Slutsky
Robins, Schwartz, Nicholas, Lifton,
Taylor, LTD.
Chicago IL 60016

Ralph Sullons
2525 Conklin Drive #5
Rockford IL 61103

Fred Tarnow
Helen Keller Junior High School
820 Bode Road
Schaumburg IL 60194

John Tofano
Elk Grove School District #59
2121 S. Arlington Heights Rd.
Arlington Heights IL 60005

SPEAKERS

Joanne Troutner
Tipecanoe School Corporation
3310 North 300 West
West Lafayette IN 47906

Gary Tubb
University of South Florida
4202 Fowler Ave.
HMS 301
Tampa FL 33620

Sandra Turner
National College of Education
2840 Sheridan Road
Evanston IL 60201

Jerry Turry
Niles West High School
Oakton St. at Edens Highway
Skokie IL 60077

Cynthia Vance
LENS, International
4760 N. Sheridan
Chicago IL 60640

Dan Watt
Educational Alternatives
Gregg Lake Road
Antrim NH 03440

Molly Watt
Educational Alternatives
Gregg Lake Road
Antrim NH 03440

Catherine Wozniak
Glenbrook South HS
4000 W. Lake Ave.
Glenview IL 60025

Gerry Zeller
Avoca School District #37
2921 Illinois Road
Wilmette IL 60091